

# Trabajo Fin de Máster Máster Ingeniería Industrial

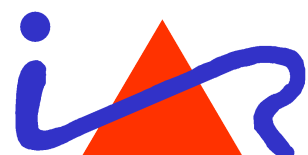
## Detección y control tolerante a fallos de planta solar

Autor: Jesús David Gordón Ballester

Tutor: Francisco Rodríguez Rubio

**Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla**

Sevilla, 2018





Trabajo Fin de Máster  
Máster Ingeniería Industrial

# **Detección y control tolerante a fallos de planta solar**

Autor:

Jesús David Gordón Ballester

Tutor:

Francisco Rodríguez Rubio

Doctor Ingeniero Industrial

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Máster: Detección y control tolerante a fallos de planta solar

Autor: Jesús David Gordón Ballester

Tutor: Francisco Rodríguez Rubio

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

Agradezco de nuevo a mi tutor Francisco Rodríguez Rubio por confiar en mí para continuar con este trabajo, y a Manuel Garrido por su gran ayuda durante el desarrollo de este trabajo. Quiero dar las gracias también a mi jefe de trabajo Juanjo, por permitirme reunirme con mi tutor o Manuel Garrido cuando tenía que ir por las mañanas a resolver dudas. Agradezco también a mi familia el hecho de respetarme el tiempo que he estado haciendo este trabajo y por último a mis compañeros, de los cuales me llevo los mejores recuerdos del paso por esta Escuela.

*Jesús David Gordón Ballester*

*Sevilla, 2018*





# Resumen

---

**E**n este trabajo se ha realizado la identificación de modelos lineales de una planta solar representado por ecuaciones no lineales. Con la identificación lineal obtenida se realiza un control predictivo basado en modelo y un control adaptativo. Por último, se ha construido un sistema de identificación de fallos basado en observadores de Luenberger, permitiendo la realización de un control tolerante a fallos de una planta solar.



# Abstract

---

In this work the identification of linear models of a solar plant represented by non-linear equations has been carried out. With the obtained linear identification a predictive control is performed by using MPC (Model Predictive Control) and an adaptative control. Finally, a fault identification system based on Luenberger observers has been built, allowing a fault-tolerant control of the solar plant.



# Índice Abreviado

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<b>1 Introducción</b>	<b>1</b>
<b>2 Descripción de la planta</b>	<b>3</b>
2.1 Captadores solares	5
2.2 Acumuladores solares	6
2.3 Acumulador de inercia	9
2.4 Caldera	9
2.5 Máquina de absorción	9
2.6 Torre de refrigeración	11
2.7 Bomba de calor	11
2.8 Intercambiador de calor	11
2.9 Bombas	11
2.10 Válvula de tres vías	12
<b>3 Identificación usando mínimos cuadrados recursivos</b>	<b>15</b>
3.1 El método de los mínimos cuadrados	16
3.2 Algoritmo recursivo para identificación en línea	18
3.3 Sistema a identificar	20
3.4 Implementación en Matlab	21
<b>4 Controlador Predictivo basado en modelo</b>	<b>27</b>
4.1 Introducción	27
4.2 Modelo	28
4.3 Control predictivo en una ventana de optimización	29
4.4 Optimización	31
4.5 Implementación en simulink	33
4.6 PID adaptativo	38

<b>5</b>	<b>Identificación de fallos</b>	<b>41</b>
5.1	Conceptos fundamentales de un observador	41
5.2	Observadores de Estado para Sistemas Lineales	42
5.3	Observadores de estado para Detección y Diagnóstico de estados	44
<b>6</b>	<b>Control tolerante a fallos</b>	<b>51</b>
6.1	Conclusiones	57
	<i>Índice de Figuras</i>	59
	<i>Índice de Tablas</i>	61
	<i>Índice de Códigos</i>	63
	<i>Bibliografía</i>	65
	<i>Índice alfabético</i>	67
	<i>Glosario</i>	67

# Índice

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<b>1 Introducción</b>	<b>1</b>
<b>2 Descripción de la planta</b>	<b>3</b>
2.1 Captadores solares	5
2.1.1 Modelo no lineal	5
2.2 Acumuladores solares	6
2.2.1 Modelo no lineal	7
2.3 Acumulador de inercia	9
2.4 Caldera	9
2.5 Máquina de absorción	9
2.5.1 Modelo no lineal del generador	9
2.6 Torre de refrigeración	11
2.7 Bomba de calor	11
2.8 Intercambiador de calor	11
2.9 Bombas	11
2.10 Válvula de tres vías	12
<b>3 Identificación usando mínimos cuadrados recursivos</b>	<b>15</b>
3.1 El método de los mínimos cuadrados	16
3.2 Algoritmo recursivo para identificación en línea	18
3.3 Sistema a identificar	20
3.4 Implementación en Matlab	21
<b>4 Controlador Predictivo basado en modelo</b>	<b>27</b>
4.1 Introducción	27
4.2 Modelo	28
4.3 Control predictivo en una ventana de optimización	29
4.3.1 Predicción de las variables de estado y la salida	29

4.4	Optimización	31
4.4.1	Restricciones	31
4.5	Implementación en simulink	33
4.6	PID adaptativo	38
<b>5</b>	<b>Identificación de fallos</b>	<b>41</b>
5.1	Conceptos fundamentales de un observador	41
5.2	Observadores de Estado para Sistemas Lineales	42
5.3	Observadores de estado para Detección y Diagnóstico de estados	44
<b>6</b>	<b>Control tolerante a fallos</b>	<b>51</b>
6.1	Conclusiones	57
	<i>Índice de Figuras</i>	59
	<i>Índice de Tablas</i>	61
	<i>Índice de Códigos</i>	63
	<i>Bibliografía</i>	65
	<i>Índice alfabético</i>	67
	<i>Glosario</i>	67



# Notación

---

$\mathbb{R}$	Cuerpo de los números reales
$\mathbb{C}$	Cuerpo de los números complejos
$\ \mathbf{v}\ $	Norma del vector $\mathbf{v}$
$\langle \mathbf{v}, \mathbf{w} \rangle$	Producto escalar de los vectores $\mathbf{v}$ y $\mathbf{w}$
$ \mathbf{A} $	Determinante de la matriz cuadrada $\mathbf{A}$
$\det(\mathbf{A})$	Determinante de la matriz (cuadrada) $\mathbf{A}$
$\mathbf{A}^\top$	Transpuesto de $\mathbf{A}$
$\mathbf{A}^{-1}$	Inversa de la matriz $\mathbf{A}$
$\mathbf{A}^\dagger$	Matriz pseudoinversa de la matriz $\mathbf{A}$
$\mathbf{A}^H$	Transpuesto y conjugado de $\mathbf{A}$
$\mathbf{A}^*$	Conjugado
c.t.p.	En casi todos los puntos
c.q.d.	Como queríamos demostrar
■	Como queríamos demostrar
□	Fin de la solución
e.o.c.	En cualquier otro caso
$e$	número $e$
$e^{jx}$	Exponencial compleja
$e^{j2\pi x}$	Exponencial compleja con $2\pi$
$e^{-jx}$	Exponencial compleja negativa
$e^{-j2\pi x}$	Exponencial compleja negativa con $2\pi$
$\text{Re}$	Parte real
$\text{Im}$	Parte imaginaria
$\text{sen}$	Función seno
$\text{tg}$	Función tangente
$\text{arc tg}$	Función arco tangente
$\sin^y x$	Función seno de $x$ elevado a $y$
$\cos^y x$	Función coseno de $x$ elevado a $y$
$\text{Sa}$	Función sampling
$\text{sgn}$	Función signo
$\text{rect}$	Función rectángulo
$\text{Sinc}$	Función sinc
$\frac{\partial y}{\partial x}$	Derivada parcial de $y$ respecto a $x$

$x^\circ$	Notación de grado, $x$ grados.
$\Pr(A)$	Probabilidad del suceso $A$
$E[X]$	Valor esperado de la variable aleatoria $X$
$\sigma_X^2$	Varianza de la variable aleatoria $X$
$\sim f_X(x)$	Distribuido siguiendo la función densidad de probabilidad $f_X(x)$
$\mathcal{N}(m_X, \sigma_X^2)$	Distribución gaussiana para la variable aleatoria $X$ , de media $m_X$ y varianza $\sigma_X^2$
$\mathbf{I}_n$	Matriz identidad de dimensión $n$
$\text{diag}(\mathbf{x})$	Matriz diagonal a partir del vector $\mathbf{x}$
$\text{diag}(\mathbf{A})$	Vector diagonal de la matriz $\mathbf{A}$
SNR	Signal-to-noise ratio
MSE	Minimum square error
:	Tal que
$\stackrel{\text{def}}{=}$	Igual por definición
$\ \mathbf{x}\ $	Norma-2 del vector $\mathbf{x}$
$ \mathbf{A} $	Cardinal, número de elementos del conjunto $\mathbf{A}$
$\mathbf{x}_i, i = 1, 2, \dots, n$	Elementos $i$ , de 1 a $n$ , del vector $\mathbf{x}$
$dx$	Diferencial de $x$
$\leq$	Menor o igual
$\geq$	Mayor o igual
$\backslash$	Backslash
$\Leftrightarrow$	Si y sólo si
$x = a + 3 \underset{\substack{\uparrow \\ a=1}}{=} 4$	Igual con explicación
$\frac{a}{b}$	Fracción con estilo pequeño, $a/b$
$\Delta$	Incremento
$b \cdot 10^a$	Formato científico
$\xrightarrow{x}$	Tiende, con $x$
$\mathbf{O}$	Orden
$\text{TM}$	Trade Mark
$\mathbb{E}[x]$	Esperanza matemática de $x$
$\mathbf{C}_x$	Matriz de covarianza de $\mathbf{x}$
$\mathbf{R}_x$	Matriz de correlación de $\mathbf{x}$
$\sigma_x^2$	Varianza de $x$

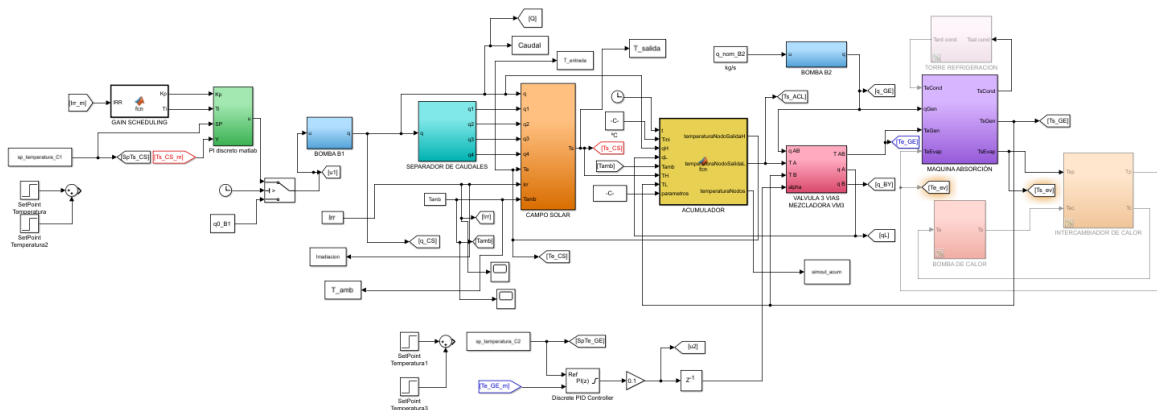
# 1 Introducción

*Grandes descubrimientos y mejoras implican invariablemente la cooperación de muchas mentes.*

ALEXANDER GRAHAM BELL

El objetivo de este trabajo es la identificación de modelos lineales a partir de ecuaciones no lineales de una planta solar de refrigeración por absorción y su control. Con los modelos obtenidos se implementa un control tanto predictivo como adaptativo y se realiza un sistema de detección de fallos con el fin de poder realizar un control tolerante a fallos.

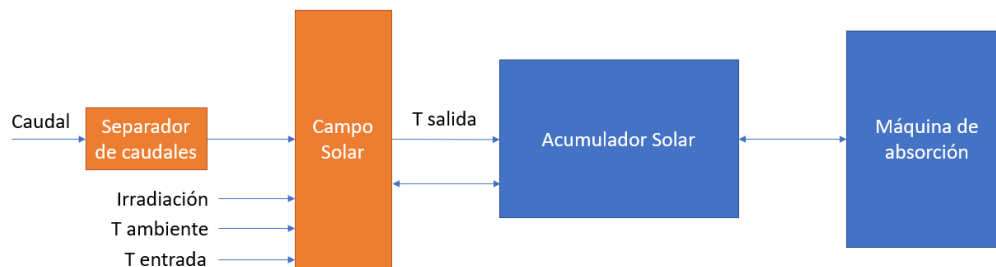
En primer lugar se hará una descripción de la planta solar, detallando cada una de las partes que componen el sistema y explicando los modelos no lineales obtenidos [2]. Éstos modelos no lineales compondrán el simulador de la planta solar. En la Figura 1.1 se puede ver el diagrama de bloques de simulink del modelo no lineal de la planta.



**Figura 1.1** Simulador de la planta solar.

Un esquema del simulador se puede ver en Figura 1.2 para su mejor entendimiento. Al sistema de colectores llegará un caudal procedente de una bomba, este caudal será la variable de entrada de nuestro sistema. El caudal pasa por un separador de caudal debido a que hay cuatro placas solares en el campo solar. Al campo solar le llegan como entradas

la temperatura ambiente, la irradiación y una temperatura de los caudales de entrada. El campo solar está interrelacionado con el acumulador y la máquina de absorción. Este trabajo se centrará en la identificación y control del campo solar.



**Figura 1.2** Esquema del simulador de la planta solar.

En segundo lugar, se llevará a cabo la identificación de modelos lineales mediante un algoritmo recursivo de identificación en línea. Se identificarán ecuaciones lineales de 2º orden del campo solar, pudiendo así, realizar posteriormente un control predictivo y adaptativo de la temperatura de salida del campo.

A continuación, se implementarán técnicas de control adaptativo y predictivo. Como control predictivo se implementará un controlador predictivo basado en modelo. El modelo será representado en el espacio de estados y será el identificado en línea, es decir, con los datos que se van obteniendo durante la simulación. El control adaptativo empleado será el de un PID en el que los parámetros irán cambiando según el modelo identificado. La variable de control será el caudal de entrada del campo de colectores y la variable a controlar será la temperatura de salida.

Por último, se ha realizado un sistema de identificación de fallos, basado en observadores de Luenberger. Se aprovechan los modelos obtenidos en línea para detectar cambios bruscos de los parámetros, identificando de esta manera los fallos del sistema. Una vez construido el sistema de identificación de fallos se realizará el control de la planta solar provocando fallos, para comprobar su eficacia.

## 2 Descripción de la planta

---

*Los científicos estudian el mundo tal como es; los ingenieros crean el mundo que nunca ha sido.*

THEODORE VON KARMAN

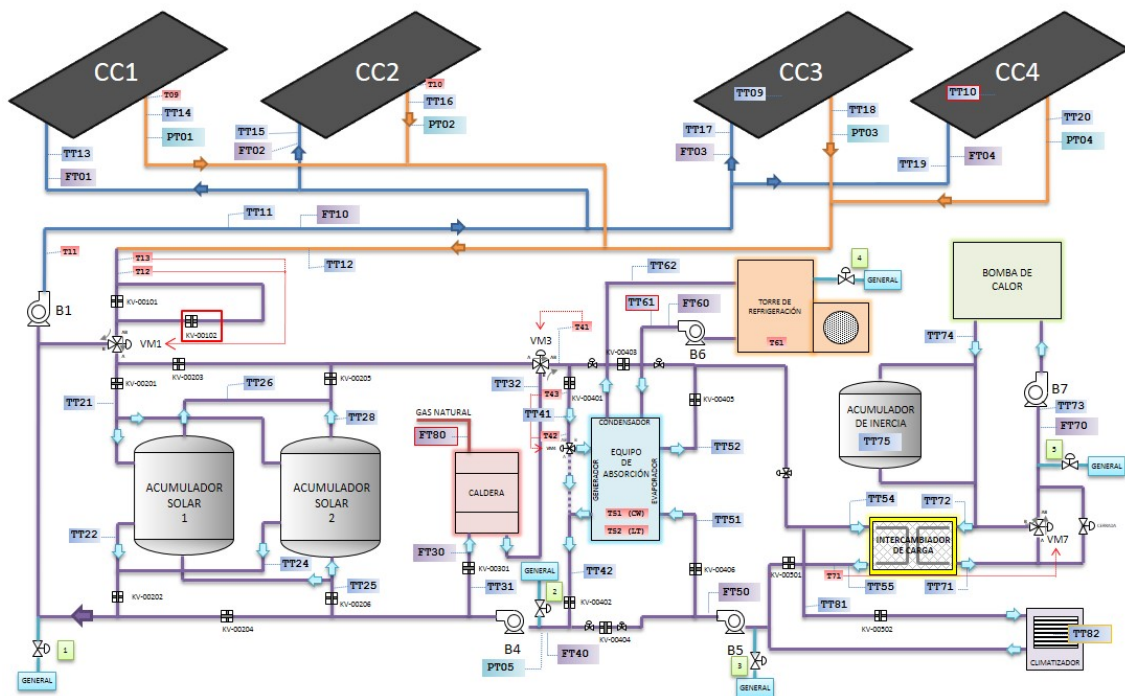
La instalación sobre la que se ha basado el proyecto fue construida en 1998 en la Escuela Superior de Ingenieros de Sevilla, en el marco de colaboración entre el Instituto Andaluz de Automática y Robótica y el Instituto Andaluz de Energías Renovables. Se trata de una instalación flexible con el fin de ir incorporando las modificaciones que se consideren necesarias para el mejor desarrollo de la tecnología de la tecnología de refrigeración solar por absorción.

La instalación consta de un equipo de absorción con una potencia frigorífica nominal de 35 KW, junto con un sistema de aporte de energía térmica necesario para el funcionamiento del ciclo de absorción y un sistema de evacuación de calor. La fuente de calor está formada por un conjunto de captadores solares térmicos que transforman la energía incidente en forma de radiación en energía interna del fluido caloportador que, en este caso, es agua. En las condiciones nominales que posteriormente se describen, la energía térmica suministrada por el sistema de captación es de 50 KW.

Como complemento al sistema de captadores existe un sistema de energía auxiliar constituido por una caldera de gas natural de 60 KW de potencia nominal para usar en aquellos casos en los que el aporte de radiación solar sea insuficiente. La energía disipada por el sistema de refrigeración por absorción es evacuada mediante una torre de refrigeración.

La demanda de energía térmica puede provenir de dos fuentes: la instalación de climatización de la planta 2 del edificio de laboratorios o un sistema simulador de cargas que se diseña para ensayos con cualquier tipo de cargas, tanto constantes para establecer condiciones estacionarias de funcionamiento como variables simulando las cargas térmicas de un edificio. Alternativamente se puede elegir entre una configuración u otra según el tipo de experimento que se quiera realizar.

A continuación, se muestra la distribución actual de la planta.



**Figura 2.1** Esquema de la planta.

La instalación permite la producción de calor y frío para climatización a partir de energía obtenida en captadores solares térmicos con el apoyo de una caldera de calor.

La planta en condiciones normales tiene 4 circuitos de agua diferentes que llegan a la máquina de absorción: el de generación de calor (campo, acumuladores y caldera que pasan por el generador de la máquina de absorción), el de generación de frío (pasa por el evaporador y va o al intercambiador o al climatizador, dependiendo si se está trabajando con simulación de carga o con carga real), el circuito de condensación (que pasa por el condensador de la máquina de absorción y va a la torre de refrigeración) y el circuito de simulación de carga (en el secundario del intercambiador).

La instalación está constituida por los siguientes sistemas que a continuación se describen brevemente:

- Captadores solares
- Acumuladores solares
- Acumulador de inercia
- Caldera
- Máquina de absorción
- Torre de refrigeración
- Bomba de calor
- Intercambiador de calor

En [1] se podrá encontrar una información más detallada de la planta.

## 2.1 Captadores solares

El sistema de captadores solares está constituido por cuatro campos: CC1, CC2, CC3 Y CC4. Cada campo de colectores (CC) está constituido por 18 paneles, dispuesto en tres baterías con 6 paneles en serie/paralelo. En condiciones nominales de funcionamiento, cada panel puede aportar al líquido que circula por su interior un calor de 50 KJ. Por tanto, el campo en su totalidad es capaz de aportar en condiciones nominales 3600 KJ.



**Figura 2.2** Campo Solar.

### 2.1.1 Modelo no lineal

Se ha utilizado un modelo de parámetros distribuidos en la dirección del flujo que aglutina todas las capacidades de los distintos materiales en una única capacidad térmica de todo el colector. Aplicando un balance de energía a un elemento diferencial de fluido en la dirección del flujo se obtiene la siguiente ecuación en derivadas parciales:

$$WC_T \frac{\partial T f}{\partial t} = F'W(I_G - U_L(T f - T_{amb})) - \dot{m}C_p \frac{\partial T f}{\partial x} \quad (2.1)$$

Para resolver la ecuación (2.1) se discretiza el espacio (la longitud del colector) en N nodos mediante (2.2) obteniendo así un modelo consistente en N ecuaciones diferenciales dadas por (2.3), cada una asociada a un nodo  $j$ ,  $j = 1, \dots, N$ .

$$\frac{\partial T f_j}{\partial x} \approx \frac{\Delta T f_j}{\Delta x} = \frac{\Delta T f_j - \Delta T f_{j-1}}{\Delta x} \quad (2.2)$$

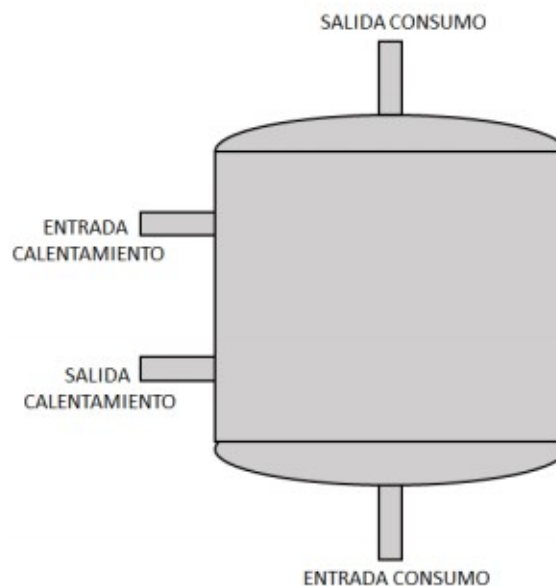
$$WC_T \frac{\partial T_f}{\partial t} = F'W(I_G - U_L(T_f - T_{amb})) - \dot{m}C_p \frac{\Delta T_f}{\Delta x} \quad (2.3)$$

**Tabla 2.1** Nomenclatura del modelo del colector.

Variable	Descripción
$T_f(K)$	Temperatura del fluido
$F'$	Factor de eficiencia del captador solar
$U_L(W/m^2K)$	Coefficiente global de pérdidas en el ambiente
$C_T(J/m^2K)$	Capacidad térmica por unidad de área del colector
$W(m)$	Ancho del captador solar
$\dot{m}(Kg/s)$	Caudal másico
$I_G(W/m^2)$	Irradiancia
$T_{amb}(K)$	Temperatura ambiente
$C_p(J/KgK)$	Calor específico del fluido

## 2.2 Acumuladores solares

Se tienen dos acumuladores solares para la acumulación del agua caliente. Cada depósito tiene una capacidad de  $2,5 m^3$ . Los acumuladores tienen el siguiente esquema.

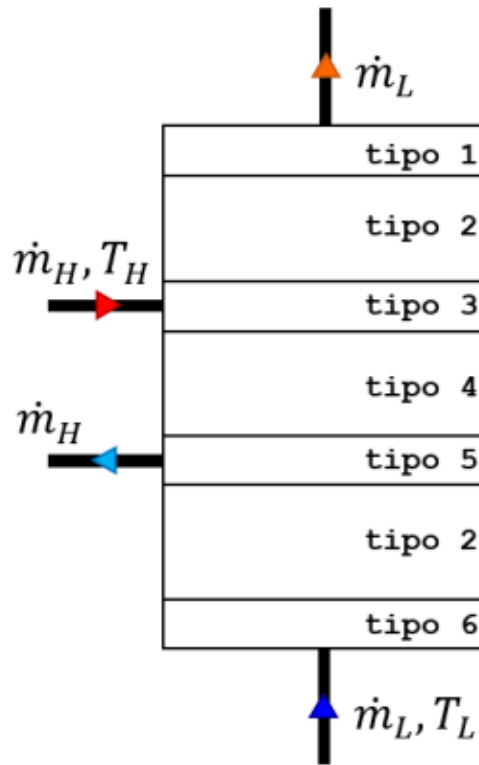
**Figura 2.3** Esquema del acumulador.



### 2.2.1 Modelo no lineal

Para el tanque acumulador también se ha elegido un modelo de parámetros distribuidos. Se divide el tanque en  $N$  segmentos o nodos y se plantea un balance de energía para cada nodo para construir un sistema de  $N$  ecuaciones diferenciales de cuya resolución se obtienen las temperaturas de cada nodo en función del tiempo. Se considera la hipótesis de que los flujos de fluido que circulan hacia arriba y hacia abajo en cada nodo se mezclan antes de entrar en cada nodo. El número de nodos se ha establecido en  $N = 24$ . La nomenclatura utilizada en el modelo del tanque acumulador se explica en la Tabla 2.2.

Podemos distinguir seis tipos de nodo con diferentes balances de energía Figura 2.4:



**Figura 2.4** Modelo de parámetros distribuidos del acumulador.

- Tipo 1: Nodo de salida del lado de la carga (generador de la máquina de absorción).

$$M_i C_p \frac{dT_i}{dt} = \dot{m}_L C_p (T_i - T_{i+1}) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i)$$

- Tipo 2: Nodo interno A.

$$M_i C_p \frac{dT_i}{dt} = \dot{m}_L C_p (T_i - T_{i+1}) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

- Tipo 3: Nodo de entrada del lado de la fuente de calor (captadores solares):

$$M_i C_p \frac{dT_i}{dt} = \dot{m}_H C_p (T_H - T_i) - \dot{m}_L C_p (T_i - T_{i+1}) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

- Tipo 4: Nodo interno B. Si  $(\dot{m}_H - \dot{m}_L) > 0$ :

$$M_i C_p \frac{dT_i}{dt} = (\dot{m}_H - \dot{m}_L) C_p (T_{i-1} - T_i) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

Si  $(\dot{m}_H - \dot{m}_L) \leq 0$

$$M_i C_p \frac{dT_i}{dt} = (\dot{m}_H - \dot{m}_L) C_p (T_i - T_{i+1}) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

- Tipo 5: Nodo de salida del lado de la fuente de calor.

$$M_i C_p \frac{dT_i}{dt} = \dot{m}_H C_p (T_{i-1} - T_i) - \dot{m}_L C_p (T_i - T_{i+1}) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i+1 \rightarrow i}} (T_{i+1} - T_i) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

- Tipo 6: Nodo de entrada del lado de la carga.

$$M_i C_p \frac{dT_i}{dt} = \dot{m}_L C_p (T_L - T_i) - UA_{s,i} (T_i - T_{amb}) + \frac{(k + \Delta k) A_{c,i}}{\Delta x_{i-1 \rightarrow i}} (T_{i-1} - T_i)$$

Tabla 2.2 Nomenclatura del acumulador.

Variable	Descripción
$M_i(Kg)$	Masa del nodo i
$C_p(J/KgK)$	Calor específico del fluido
$U(W/m^2K)$	Coeficiente global de pérdidas al ambiente
$k(W/mK)$	Conductividad del fluido
$\Delta x(m)$	Distancia entre líneas medias de dos nodos consecutivos
$A_{s,i}(m^2)$	Área de la pared del nodo i
$A_{c,i}(m^2)$	Área de la sección del tanque
$\dot{m}_H(Kg/s)$	Caudal másico de la fuente de calor
$\dot{m}_L(Kg/s)$	Caudal másico de la carga
$T_H(K)$	Temperatura de la toma de entrada procedente de la fuente de calor
$T_L(K)$	Temperatura de la toma de entrada procedente de la carga

## 2.3 Acumulador de inercia

Se trata de un dispositivo de  $1 m^3$  de capacidad aislado con poliuretano. Su función es la de aportar agua cuando la bomba de calor no pueda atender las demandas. Este depósito evita el desgaste de la bomba de calor, ya que sin acumulador de inercia se producirían procesos continuos de arranque y parada.

## 2.4 Caldera

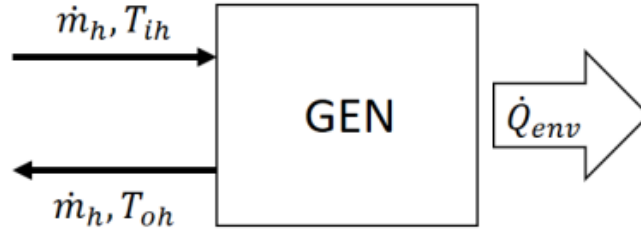
La caldera es el sistema de energía auxiliar. Se trata de una caldera de gas natural que cuenta con una potencia nominal de 60 KW. Su función es la de calentar el agua a la entrada del generador del equipo de absorción cuando el campo de colectores no pueda alcanzar la temperatura mínima de funcionamiento del generador ( $75^\circ C$ ). La temperatura máxima de trabajo de esta caldera es de  $110^\circ C$ , y cuenta con un termostato ajustable manualmente para fijar una temperatura máxima de operación.

## 2.5 Máquina de absorción

La máquina de absorción cuenta con una potencia nominal de 35 KW. Es la encargada de la producción de frío en la planta solar, ya que permite traspasar energía de una fuente a baja temperatura a otra a alta temperatura con un pequeño consumo de energía adicional.

### 2.5.1 Modelo no lineal del generador

Aplicando un balance de potencia al flujo de agua que circula por el generador se obtiene (2.4):



**Figura 2.5** Balance de potencia en el generador.

$$C_{gen} \frac{dT_{oh}}{dt} = \dot{m}_h C_p (T_{ih} - T_{oh}) - \dot{Q}_{amb} \quad (2.4)$$

Donde  $\dot{Q}_{amb}$  es la potencia cedida al ambiente por el generador (pérdidas). Este término se modela de la siguiente manera:

$$\dot{Q}_{amb} = (T_{mh} - T_{amb}) U A a_{gen}$$

Siendo  $T_{mh}$  la temperatura media del agua en el generador:

$$T_{mh} = \frac{T_{ih} + T_{oh}}{2}$$

A partir de los datos de funcionamiento nominal de la máquina de absorción se puede estimar el valor de la capacidad térmica del generador:

$$C_{gen} = \frac{\dot{Q}_{gen} t}{\Delta T} = \frac{50 \text{ kW} \cdot 1 \text{ s}}{5 \text{ K}} = 10 \text{ kJ/K}$$

donde  $\Delta T$  es la diferencia entre la temperatura de entrada y de salida del agua del generador, y  $\dot{Q}_{gen}$  es la potencia cedida al generador, la cual se obtiene de la definición del COP:

$$COP = \frac{\dot{Q}_{ev}}{\dot{Q}_{gen}}$$

**Tabla 2.3** Nomenclatura del generador.

Variable	Descripción
$T_{oh}(K)$	Temperatura del agua a la salida del generador
$C_{gen}(J/K)$	Capacidad térmica del generador
$UA_{gen}(W/K)$	Coefficiente global de pérdidas al ambiente del generador
$\dot{m}_h(Kg/s)$	Caudal másico de agua que atraviesa el generador
$T_{ih}(K)$	Temperatura del agua a la entrada del generador
$T_{mh}(K)$	Temperatura media del agua en el generador
$C_p(J/KgK)$	Temperatura media del agua en el generador

## 2.6 Torre de refrigeración

La torre de refrigeración es el también llamado sistema de evacuación de calor. Su función fundamental es la extracción del calor de la corriente de agua que llega de la máquina de absorción. Posee un motor de potencia de 1.1 KW. La torre tiene una potencia nominal de 80 KW, con un rango de temperaturas de entrada desde los 25 a los 31 °C.

## 2.7 Bomba de calor

La bomba de calor realiza las funciones de simulador de carga junto con el acumulador de inercia. La bomba de calor es capaz de invertir su funcionamiento para producir calor además de frío. La bomba de calor funcionará normalmente en modo producción de calor, ya que la instalación se usará normalmente en el modo de producir frío. El equipo instalado cuenta con potencia nominal de 55.4 KW para la producción de calor, y 51.4 KW para frío.

## 2.8 Intercambiador de calor

Se trata de un intercambiador agua-agua que permite simular cargas variables mediante una válvula de tres vías controlada, VM7. Estas cargas serán de 0 a 50 kW. El caudal de circulación es constante, por lo que la demanda de energía queda definida por las variaciones de temperatura.

## 2.9 Bombas

Las bombas se modelarán como un sistema de primer orden de ganancia unidad:

$$\tau \frac{dq_o}{dt} = q_i - q_o$$

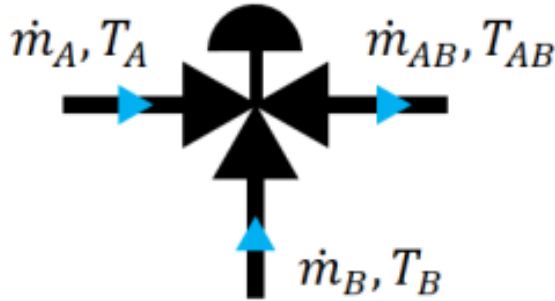
Donde  $q_i$  es el caudal deseado,  $q_o$  es el caudal que proporciona la bomba y  $\tau$  es la constante de tiempo. El modelo incluye una saturación para que el caudal no pueda hacerse negativo ni tampoco superar un caudal límite,  $q_{sat}$ . Esta saturación se aplica a la variable  $q_i$ .

## 2.10 Válvula de tres vías

La válvula de tres vías actúa como mezcladora. El factor de apertura de la válvula,  $\alpha$ , varía entre cero y uno regulando la proporción de caudales A y B de manera lineal.

$$\alpha = 0 \rightarrow \dot{m}_{AB} = \dot{m}_B$$

$$\alpha = 1 \rightarrow \dot{m}_{AB} = \dot{m}_A$$



**Figura 2.6** Válvula de tres vías.

Las entradas del modelo son el factor de apertura deseado  $\alpha_d$ , el caudal  $q_{AB}$ , y la temperatura de las tomas A y B. La salida es la temperatura de la toma AB. Aplicando un balance de potencia sobre el elemento se obtiene la siguiente ecuación:

$$\dot{m}_A C_p T_A + \dot{m}_B C_p T_B = \dot{m}_{AB} C_p T_{AB}$$

Dado que:

$$\dot{m}_A = \alpha \dot{m}_{AB}$$

$$\dot{m}_B = (1 - \alpha)\dot{m}_{AB}$$

La temperatura a la salida toma la siguiente expresión:

$$\begin{aligned} T_{AB} &= (\alpha\dot{m}_{AB}T_A + (1 - \alpha)\dot{m}_{AB}T_B)/\dot{m}_{AB} \\ &= \alpha T_A + (1 - \alpha)T_B \end{aligned}$$

Se distingue en el modelo el factor de apertura deseado,  $\alpha_d$ , del factor de apertura real de la válvula,  $\alpha$ . Al factor de apertura se le aplica una limitación en su tasa de cambio (Sr). Si la válvula tarda  $n$  segundos en pasar de la posición B ( $\alpha=0$ ) a la A ( $\alpha=1$ ), la tasa de cambio es  $1/n$ .





# 3 Identificación usando mínimos cuadrados recursivos

---

*No me importa que hayan robado mi idea. Me importa que no tengan ninguna propia.*

NIKOLA TESLA

La *identificación de sistemas* es uno de los problemas más complejos que se pueden dar. La identificación correcta de un sistema abre muchas posibilidades: una correcta predicción de los valores de un sistema puede servir de prevención, evitando por ejemplo desgaste de actuadores, deterioro de partes del sistema por temperaturas extremas, etc. También puede ayudar a construir un buen detector de fallos, siendo más fiable cuanto mayor es la precisión del modelo construido. Si el modelo es preciso, se puede llegar a controlar dicho sistema tal y cómo se quiere, lo que hace que tener una buena identificación sea una herramienta muy poderosa.

Hay distintas formas de identificar un sistema:

- Identificación off-line
- Identificación on-line

La identificación off-line es aquella que se realiza con una serie de datos recogidos del sistema tal y cómo se hizo en [1]. Se toman los datos de la planta real para después ajustar el modelo. Este tipo de procedimientos suelen obtener modelos más precisos y son más fiables en cuanto a la convergencia de los parámetros estimados a los parámetros reales del proceso. Como desventaja, no tienen en cuenta que los sistemas con el tiempo pueden modificar su dinámica, cambiando muchos de sus parámetros, y por lo tanto, a largo plazo dependiendo del sistema, puede llegar a dar un modelo no del todo preciso.

En cuanto a los métodos basados en línea (identificación on-line), la estimación se efectúa usando medidas que se van obteniendo en tiempo real, y normalmente se usan cálculos recursivos. Este método suele ser el único apropiado cuando se pretende usar una estrategia de control adaptativo, o el proceso varía su dinámica con el tiempo.

### 3.1 El método de los mínimos cuadrados

Tal y como se describe en [6] y [9], este método permite la identificación en tiempo real de modelos con el único requisito de que estos sean lineales en los parámetros. Esto incluye, por tanto, a modelos lineales y no lineales que sean lineales en los parámetros. El mayor interés práctico reside, sin embargo, en la identificación de los primeros, dado que son los más utilizados en control.

Considerése el siguiente modelo paramétrico lineal monovariante:

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-1) + \dots + b_n u(k-n) \quad (3.1)$$

Nótese que este modelo es determinista en el sentido de que no considera ruidos aleatorios. Es inmediato comprobar que este modelo corresponde a la siguiente función de transferencia:

$$G(z) = \frac{b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

El modelo (3.1) se puede reescribir como:

$$y(k) = m(k)\theta \quad (3.2)$$

donde el vector

$$m(k) = [-y(k-1) \dots -y(k-n) \ u(k-1) \dots u(k-n)] \quad (3.3)$$

es llamado regresor y

$$\theta = [a_1 \dots a_n \ b_1 \dots b_n]^T$$

es el vector de parámetros. Dado un valor del vector de parámetros  $\hat{\theta}$ , el error de predicción para el instante  $k$  será:

$$e(k, \hat{\theta}) = y(k) - \hat{y}(k) = y(k) - m(k)\hat{\theta}$$

Nótese que conocido el valor de los valores presentes y pasados de la salida y la entrada, la expresión (3.2) es una ecuación en la que las  $2n$  incógnitas son los parámetros que forman  $\theta$ . Si el proceso a identificar correspondiese exactamente con un modelo como (3.1) se podría determinar el valor del vector de parámetros a partir de  $2n$  medidas u observaciones de la salida para una serie de entradas conocidas. Es decir, se formaría un sistema de  $2n$  ecuaciones con el que se podría determinar el valor «real» de  $\theta$ .

El método de los mínimos cuadrados parte de  $N$  pares  $(y(k), m(k))$  donde  $N$  es generalmente mucho mayor de  $2n$  (este sería el conjunto de estimación) y permite ajustar un modelo del tipo (3.1). En el supuesto poco realista de que el proceso coincida con un modelo como el que se intenta ajustar, se tendría solución y el error de predicción alcanzado sería cero para todas las medidas del conjunto de estimación. Sin embargo, en la práctica el proceso no se puede describir a la perfección mediante un modelo lineal del tipo (3.1) por lo que el sistema de ecuaciones no tiene solución en el sentido de que no existe un valor del vector de parámetros que haga que el error de predicción sea cero para todas las medidas del conjunto de estimación. Es decir, el sistema de ecuaciones es incompatible. Sin embargo si se puede encontrar un valor del vector de parámetros que haga mínimo el error de predicción, de manera más precisa que haga mínima la suma de los cuadrados de los errores de predicción del conjunto de estimación. Esta es precisamente la estrategia del método de mínimos cuadrados.

Las medidas obtenidas desde  $k = n$  hasta  $k = N$  se agrupan en vectores de manera que se obtiene:

$$E(N, \theta) = Y(N) - M(N)\theta$$

donde los vectores  $E(N)$  e  $Y(N)$  son

$$E(N, \theta) = [e(n, \theta) \cdots e(N, \theta)]^T$$

$$Y(N) = [y(n) \cdots y(N)]^T$$

y la matriz  $M(N)$  está formada por los regresores correspondientes, es decir

$$M = \begin{bmatrix} m(n) \\ \vdots \\ m(N) \end{bmatrix}$$

Se define el índice de bondad de ajuste como

$$J(\theta) = \|E(N, \theta)\|^2 = \left( \sum_{k=n}^N e^2(k, \theta) \right)$$

Este índice lo podemos reescribir como

$$J(\theta) = (Y(N) - M(N)\theta)^T (Y(N) - M(N)\theta)$$

El mínimo valor de  $J(\theta)$  se dará en el valor del vector de parámetros que cumpla que

$$\frac{dJ(\theta)}{d\theta} = 0$$

es decir,

$$2(M(N)\theta - Y(N))^T M(N) = 0$$

de donde se obtiene que el valor del vector de parámetros que hace mínimo el índice de bondad de ajuste es:

$$\theta^* = [M^T(N)M(N)]^{-1}M^T(N)Y(N) \quad (3.4)$$

y ese es por tanto el valor del vector de parámetros del modelo identificado.

Nótese que para que el problema de identificación tenga solución la matriz  $[M^T(N)M(N)]$  tiene que ser invertible al igual que  $M(N)$ . Sin entrar en demasiados detalles, tal condición se verifica cuando la entrada cumple las condiciones de excitación persistente del sistema. Se deberá acudir por tanto a señales de entrada parecidas al ruido blanco.

### 3.2 Algoritmo recursivo para identificación en línea

La expresión (3.4) implica la inversión de una matriz que puede tener unas dimensiones apreciables. Por tanto, este algoritmo se destina a la identificación fuera de línea. En línea se emplea otro procedimiento que se muestra a continuación.

La estimación para el instante  $k$  usando las medidas obtenidas desde el instante  $n$  vendrá dada por

$$\hat{\theta} = [M^T(k)M(k)]^{-1}M^T(k)Y(k) \quad (3.5)$$

$$= P(k)M^T(k)Y(k) \quad (3.6)$$

$$= P(k)(M^T(k-1)Y(k-1) + m^T(k)y(k)) \quad (3.7)$$

donde

$$P(k) = [M^T(k)M(k)]^{-1} = \left[ \sum_{i=n}^k m^T(i)m(i) \right]^{-1}$$

es la llamada matriz de covarianza. Se puede comprobar que

$$P^{-1}(k-1) = P^{-1}(k) - m^T(k)m(k)$$

Por otra parte también se puede obtener que:

$$\begin{aligned} M^T(k-1)Y(k-1) &= P^{-1}(k-1)\hat{\theta}(k-1) \\ &= P^{-1}(k)\hat{\theta}(k-1) - m^T(k)m(k)\hat{\theta}(k-1) \end{aligned}$$

Combinando las dos últimas expresiones con la (3.6) se obtiene

$$\hat{\theta}(k) = \theta(k-1) - P(k)m^T(k)m(k)\hat{\theta}(k-1) + P(k)m^T(k)y(k) \quad (3.8)$$

$$= \theta(k-1) + P(k)m^T(k)(y(k) - m^T(k)\hat{\theta}(k-1)) \quad (3.9)$$

$$= \theta(k-1) + K(k)(y(k) - m(k)\hat{\theta}(k-1)) \quad (3.10)$$

donde  $K(k) = P(k)m^T(k)$ . Por tanto se puede expresar en forma recursiva, es decir, en función del valor del estimador en el instante anterior más un término corrector que consiste en el error de predicción en el instante actual cometido por el estimador calculado en el instante anterior multiplicado por una ganancia de adaptación  $K(k)$ . Esta fórmula da lugar al llamado algoritmo de mínimos cuadrados recursivos, que consiste en

1. Dar valores iniciales a la matriz  $P$  y al vector de parámetros  $\theta$ .

2. En cada instante  $k$

a) Leer los valores de  $y(k)$  y  $u(k)$ .

b) Formar el vector regresor  $m(k)$  según la expresión (3.3)

c) Calcular  $P(k)$  mediante

$$P(k) = P(k-1) - \frac{P(k-1)m^T(k)m(k)P(k-1)}{1 + m(k)P(k-1)m^T(k)}$$

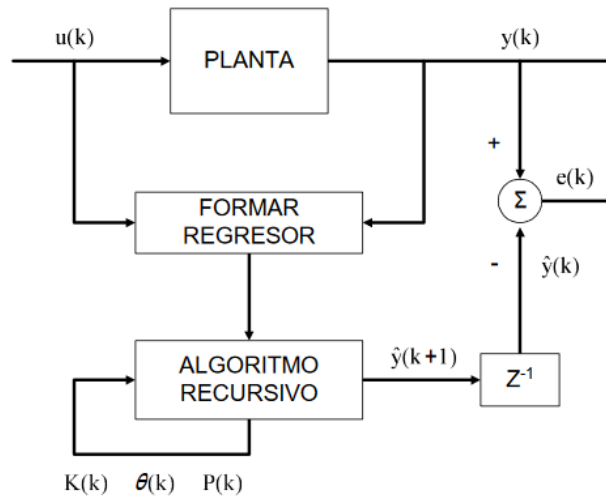
d) Calcular  $K(k)$  según la expresión

$$K(k) = P(k)m^T(k)$$

e) Calcular  $\theta(k)$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)[y(k) - m(k)\hat{\theta}(k-1)]$$

Este algoritmo se puede interpretar gráficamente como se ilustra en la figura Figura 3.1

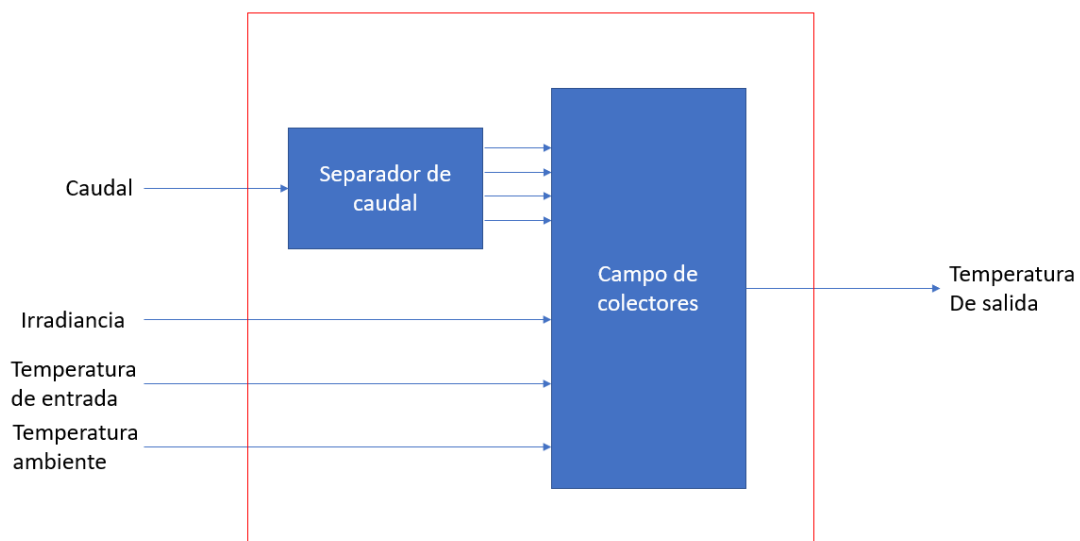


**Figura 3.1** Esquema del algoritmo.

### 3.3 Sistema a identificar

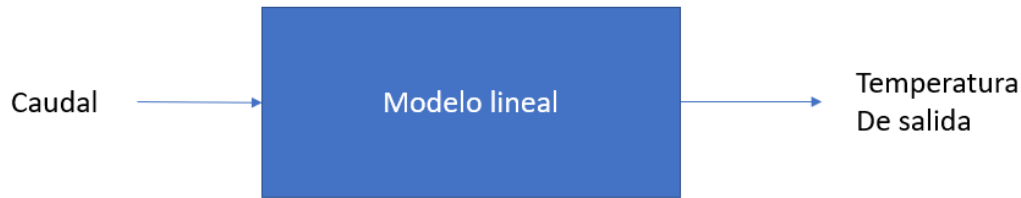
El sistema a identificar será el campo de colectores. El objetivo es realizar la identificación de un modelo lineal a partir de las ecuaciones no lineales que describen el comportamiento de la planta solar.

En el campo de colectores del simulador se tiene un sistema compuesto por un separador de caudal y al que le llegan como entradas un caudal, irradiancia, temperatura ambiente y temperatura del agua a la entrada de los colectores. La salida del sistema es la temperatura del agua a la salida. En la Figura 3.2 se puede ver las entradas y salidas del modelo no lineal.



**Figura 3.2** Entradas y salidas del modelo no lineal.

Para la identificación se obtiene un modelo lineal que relaciona el caudal, por ser la variable manipulable, con la temperatura de salida como se puede ver en la Figura 3.3.



**Figura 3.3** Entradas y salidas del modelo lineal.

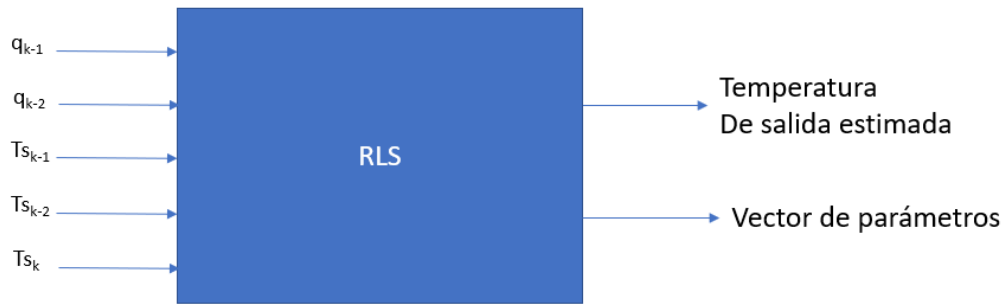
El siguiente paso es decidir el orden del modelo. Se ha optado por usar un modelo de segundo orden debido a dos motivos: por un lado recoge mayor información que un sistema de primer orden y por tanto se puede adaptar a una dinámica más compleja, y por otro lado, es menos probable que se de un sobreajuste del modelo como podría ocurrir con uno de tercer orden. Por lo tanto, el modelo a identificar será:

$$y_k = -y_{k-1}\theta_1 - y_{k-2}\theta_2 + u_{k-1}\theta_3 + u_{k-2}\theta_4$$

Siendo  $k$  el instante actual.

### 3.4 Implementación en Matlab

Para su implementación en simulink, se ha realizado una *matlab function*. Esta función tiene como argumentos de entrada el caudal y la temperatura de salida en los instantes anteriores, y la temperatura de salida en el instante actual. Como salidas, la temperatura de salida estimada y el vector de parámetros.



$q$  = Caudal de entrada en el campo de colectores del simulador

$T_s$  = Temperatura de salida del campo de colectores del simulador

**Figura 3.4** Entradas y salidas del algoritmo de identificación.

### Código 3.1 Algoritmo RLS.

```
function [Ts_estimada,theta] = fcn(yk1,yk2,uk1,uk2,yk)

%Formo el vector regresor
m=[-yk1 -yk2 uk1 uk2];

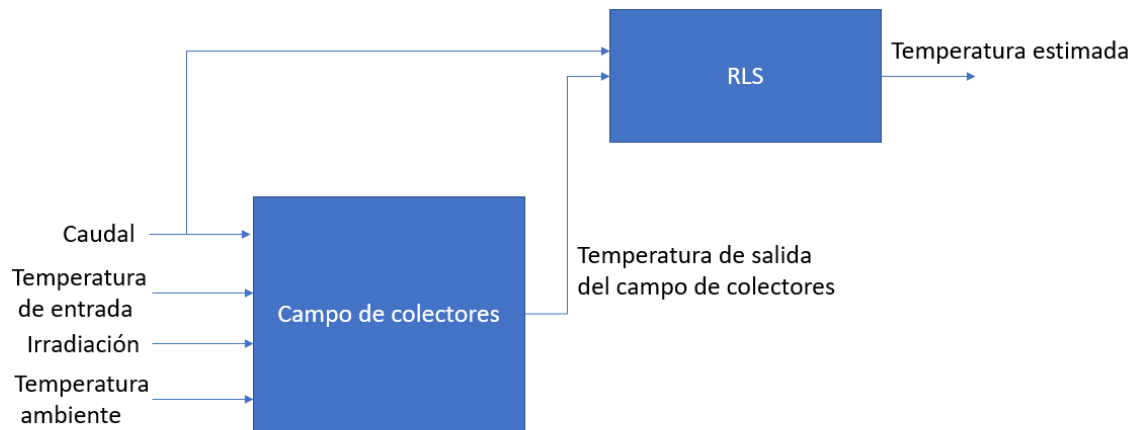
%Inicializo theta y Pk
theta=[0;0;0;0];
Pk=[30000 0 0 0;0 30000 0 0;0 0 30000 0;0 0 0 30000];

Pk=Pk-(Pk*m'*m*Pk)/(1+m*Pk*m');
K=Pk*m';
theta=theta+K*(yk-m*theta);

%Estimación
Ts_estimada=m*theta;
```

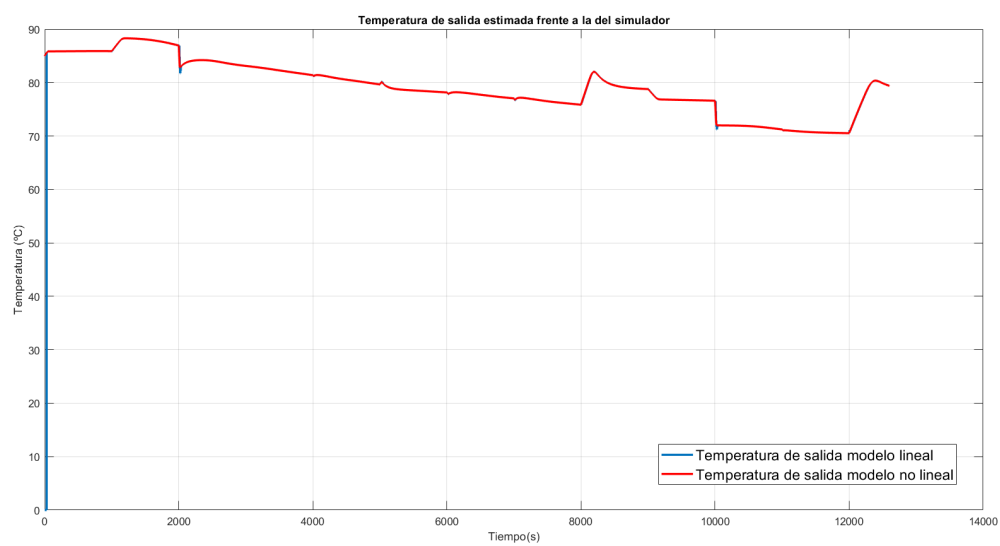
Para que el algoritmo recursivo funcionara correctamente no podía comenzar en el instante inicial de simulación, debido a que necesita un tiempo para tener valores en los instantes anteriores. El tiempo de muestreo empleado ha sido de 5 segundos. Se realizará una simulación en bucle abierto, dando una entrada de números aleatorios para comprobar que el sistema se va identificando correctamente. En la Figura 3.5 se puede ver el diagrama de bloques de la simulación realizada y en la Figura 3.8 la entrada al sistema.



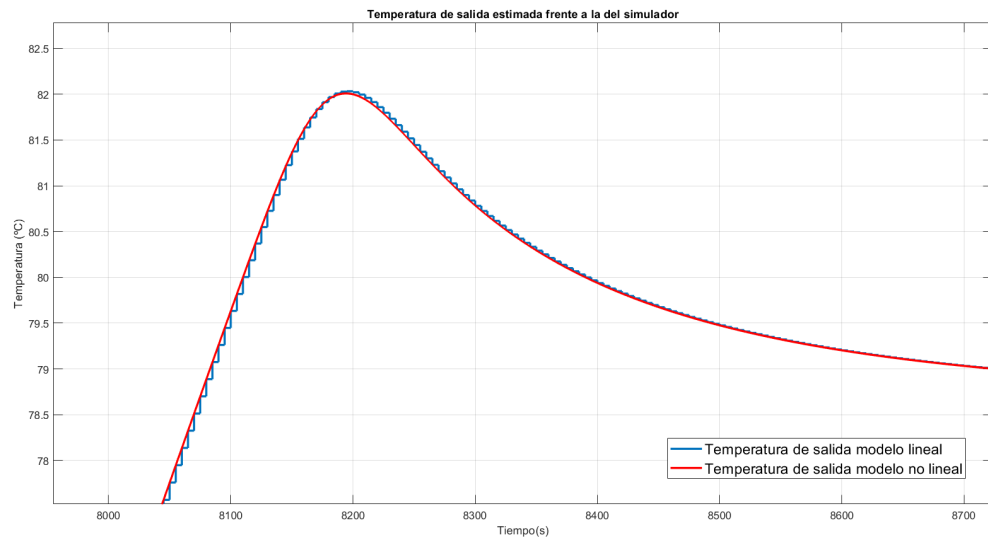


**Figura 3.5** Diagrama de bloques de la simulación realizada.

A continuación se muestra los resultados obtenidos:

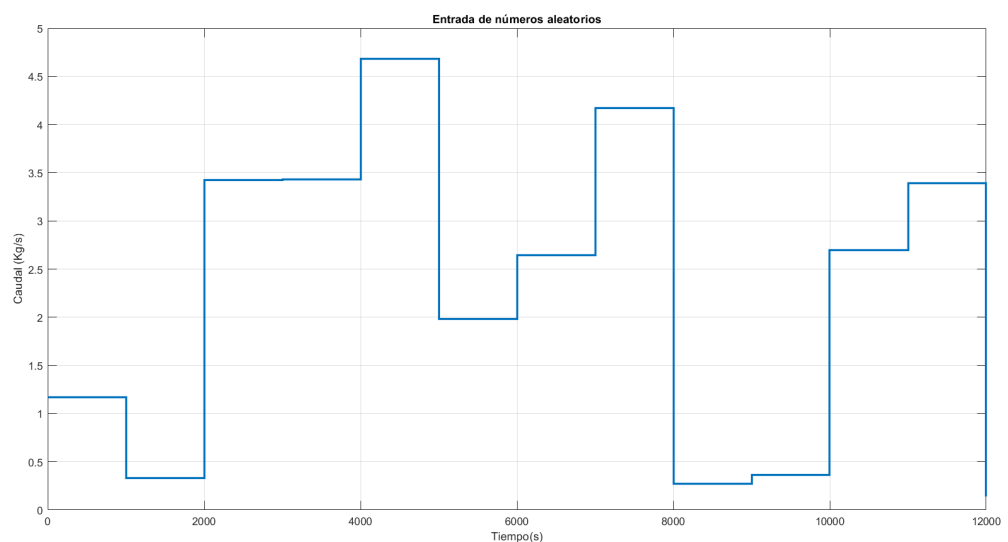


**Figura 3.6** Comparación del modelo obtenido con el simulador.



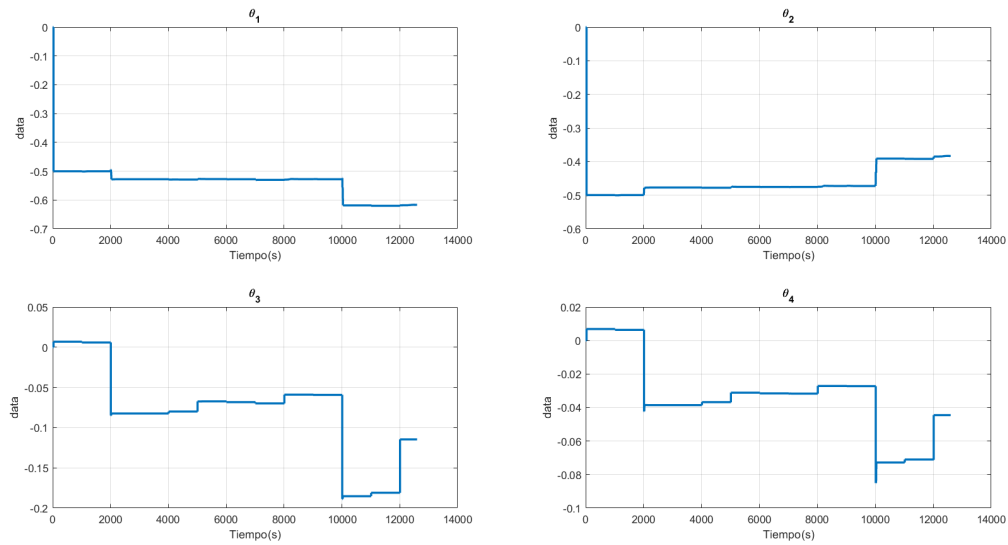
**Figura 3.7** Comparación del modelo obtenido con el simulador.

Como se ve, el modelo sigue perfectamente a la temperatura obtenida de la planta. Sólo al comienzo hay un desfase debido a la inicialización del algoritmo. En la Figura 3.8 se muestra el caudal de entrada que se le ha dado al sistema, comprendido entre los valores de saturación de la bomba (0-5 Kg/s)



**Figura 3.8** Caudal de entrada variable.

Los parámetros del sistema ajustado son los siguientes:



**Figura 3.9** Parámetros obtenidos.

Se puede apreciar cómo los parámetros van cambiando a lo largo del tiempo, aunque sin sufrir grandes variaciones. Esto se debe a que el sistema tiene una dinámica lenta y por tanto, nunca se produce un cambio brusco en la salida del sistema.



## 4 Controlador Predictivo basado en modelo

---

*Lo bueno de la ciencia es que es cierta, creas o no en ella*

NEIL DEGRASSE TYSON

**E**l control de una planta solar puede ser un tema muy complejo a tratar, debido principalmente a su no linealidad y sus grandes incertidumbres.

Este tipo de sistemas está sujeto a grandes perturbaciones como puede ser la irradiancia. Para ello existen distintas técnicas de control adaptativo como puede ser el control gain-scheduling. Usando este enfoque se permite corregir las perturbaciones que recibe el sistema de manera instantánea.

Para este trabajo se ha implementado un controlador predictivo basado en modelo. Se va aprovechar el modelo que se obtiene recursivamente para proporcionárselo al controlador y ver cómo se comporta.

### 4.1 Introducción

Los controladores predictivos basados en modelo (MPC) no son una estrategia de control específica, sino más bien se trata de un amplio rango de técnicas de control como se puede ver en [4]. Estos modelos nos llevan a controladores lineales que tienen prácticamente la misma estructura. Las ideas que aparecen en mayor o menor grado en todas las familias de controladores predictivos son las siguientes:

- Uso explícito del modelo para predecir la salida del sistema en instantes de tiempo futuro (horizonte).
- Cálculo de la acción de control minimizando una función objetivo.
- Estrategia de retroceso, en cada instante el horizonte es cambiado hacia el futuro, implementando la primera acción de control de la secuencia calculada en cada instante.

Los diferentes controladores MPC difieren entre sí en el modelo en el que son representados y en la función de coste a minimizar. En este trabajo el modelo será representado en el espacio de estados y la función a minimizar será:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U$$

Esta función estará sujeta a restricciones en la acción de control.

En [8] se explica paso a paso su implementación.

## 4.2 Modelo

Como se dijo anteriormente, el modelo utilizado será en el espacio de estados. Se considerará un sistema de una única entrada y una única salida, ya que, así son los modelos que se han identificado.

El sistema será descrito por

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (4.1)$$

$$y(k) = C_m x_m(k) \quad (4.2)$$

donde  $u$  es la variable manipulada;  $y$  es la salida del proceso y  $x_m$  es el vector de estados de dimensión  $n_1$ .

Nótese que una formulación general del espacio de estados tiene una relación directa entre la entrada y la salida

$$y(k) = C_m x_m(k) + D_m u(k)$$

Sin embargo, debido a la estrategia de retroceso, se ha asumido que la señal de entrada no podía afectar directamente a la de salida al mismo tiempo. Por lo tanto,  $D_m = 0$  en el modelo de la planta.

Tomando la diferencia con el instante anterior de (4.1) obtenemos

$$x_m(k+1) - x_m(k) = A_m(x_m(k) - x_m(k-1)) + B_m(u(k) - u(k-1))$$

Denotando la diferencia de la variable de estado por:

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k)$$

$$\Delta x_m(k) = x_m(k) - x_m(k-1)$$

y la diferencia de la variable de control por:

$$\Delta u(k) = u(k) - u(k-1)$$

Estos son los incrementos de las variables  $x_m(k)$  y  $u(k)$ . Con esta transformación, la diferencia de la ecuación de espacio de estados es:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (4.3)$$

Nótese que la entrada del modelo de espacio de estados es  $\Delta u(k)$ . El próximo paso es conectar  $\Delta x_m(k)$  con la salida  $y(k)$ . Para hacerlo, un nuevo vector variable de estado es elegido:

$$x(k) = [\Delta x_m(k)^T \ y(k)]^T$$

Restando la salida en el instante  $k+1$  y  $k$  se obtiene:

$$y(k+1) - y(k) = C_m(x_m(k+1) - x_m(k)) = C_m \Delta x_m(k+1) \quad (4.4)$$

$$= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \quad (4.5)$$

Juntando la ecuación (4.3) con (4.5) obtenemos el siguiente modelo:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (4.6)$$

$$y(k) = \begin{bmatrix} o_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (4.7)$$

donde  $o_m = [0 \ 0 \ \dots \ 0]$  de dimensión  $n_1$ . Las matrices A,B y C serán llamadas ahora matrices aumentadas.

## 4.3 Control predictivo en una ventana de optimización

Una vez que se tiene el modelo matemático, el próximo paso es calcular la predicción de la salida junto a las acciones futuras de control. Se asume que el tiempo actual es  $k_i$  y la longitud de la ventana de optimización es  $N_p$ .

### 4.3.1 Predicción de las variables de estado y la salida

Suponiendo que en el instante  $k_i$ ,  $k_i > 0$ , el vector de estados  $x(k_i)$  se puede medir, proporciona la información actual de la planta. La trayectoria futura de control es denotada por:

$$\Delta u(k_i), \Delta u(k_i+1), \dots, \Delta u(k_i+N_c-1)$$

donde  $N_c$  es llamado el control de horizonte imponiendo el número de parámetros usado para capturar la trayectoria de control futura. Dado  $x(k_i)$  se calculan los futuros vectores de estados:

$$x(k_i + 1 | k_i), x(k_i + 2 | k_i), \dots, x(k_i + m | k_i), \dots, x(k_i + N_p | k_i)$$

donde  $x(k_i + m | k_i)$  es el vector de estado predicho en  $k_i + m$  dada la actual información de la planta  $x(k_i)$ . El horizonte de control  $N_c$  es elegido para ser menor que (o igual que) el horizonte de predicción  $N_p$ . Teniendo el modelo, los futuros vectores de estado se calculan secuencialmente:

$$\begin{aligned} x(k_i + 1 | k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i + 2 | k_i) &= Ax(k_i + 1 | k_i) + B\Delta u(k_i + 1) \\ &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \end{aligned}$$

obteniendo:

$$\begin{aligned} x(k_i + N_p | k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i + 1) \\ &\quad + \dots + A^{N_p-N_c}B\Delta u(k_i + N_c - 1) \end{aligned}$$

En cuanto a la salida:

$$\begin{aligned} y(k_i + 1 | k_i) &= CAx(k_i) + CB\Delta u(k_i) \\ y(k_i + 2 | k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\ y(k_i + 3 | k_i) &= CA^3x(k_i) + CA^2B\Delta u(k_i) + CAB\Delta u(k_i + 1) \\ &\quad + CB\Delta u(k_i + 1) \end{aligned}$$

obteniendo:

$$\begin{aligned} y(k_i + N_p | k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i + 1) \\ &\quad + \dots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1) \end{aligned}$$

Definiendo los vectores

$$Y = [y(k_i + 1 | k_i) \ y(k_i + 2 | k_i) \ y(k_i + 3 | k_i) \ \dots \ y(k_i + N_p | k_i)]^T$$

$$\Delta U = [\Delta u(k_i) \ \Delta u(k_i + 1) \ \dots \ \Delta u(k_i + N_c - 1)]^T$$

Poniendo todo de forma más compacta:

$$Y = Fx(k_i) + \Phi\Delta U \tag{4.8}$$



donde

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

## 4.4 Optimización

Para una referencia dada de señal  $r(k_i)$  en un instante de tiempo  $k_i$ , en un horizonte de predicción dado, el objetivo es llevar la salida lo más cerca posible de la referencia. Este objetivo se traslada a encontrar el mejor vector  $\Delta U$  que minimice el error entre la salida y la referencia:

$$R_s^T = \overbrace{[1 \ 1 \ \dots \ 1]}^{N_p} r(k_i)$$

Se define la función objetivo:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (4.9)$$

Para resolverla, se impondrán restricciones en la acción de control, ya que, nuestra acción de control está limitada. Es posible resolver la función objetivo sin restricciones y saturar. Pero esta solución empeora los resultados del controlador.

### 4.4.1 Restricciones

Habiendo formulado las restricciones como parte del diseño, el próximo paso es trasladarlo a desigualdades lineales, y relacionarlas con el modelo original. Las restricciones son expresadas en un set de ecuaciones lineales basadas en el vector de parámetros  $\Delta U$ . Debido al horizonte de retroceso, las restricciones son consideradas en cada ventana de horizonte móvil. Si queremos imponer restricciones en la señal  $\Delta u(k)$  en el instante  $k_i$ , las restricciones son expresadas de la siguiente forma:

$$\Delta u^{min} \leq \Delta u(k_i) \leq \Delta u^{max}$$

Desde el instante  $k_i$ , el esquema de control predictivo mira hacia el futuro. Las restricciones en los instantes siguientes, por ejemplo en los tres instantes siguientes serán:

$$\begin{aligned}\Delta u^{min} &\leq \Delta u(k_i) \leq \Delta u^{max} \\ \Delta u^{min} &\leq \Delta u(k_i + 1) \leq \Delta u^{max} \\ \Delta u^{min} &\leq \Delta u(k_i + 2) \leq \Delta u^{max}\end{aligned}$$

Las restricciones en términos de  $\Delta U$  serán expresadas de la siguiente forma:

$$\Delta U^{min} \leq \Delta U \leq \Delta U^{max}$$

que serán representadas por dos desigualdades:

$$-\Delta U \leq -\Delta U^{min}$$

$$\Delta U \leq \Delta U^{max}$$

En forma matricial queda:

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

Tradicionalmente, las restricciones son impuestas para todos los instantes futuros, y todas las restricciones son expresadas en términos del vector de parámetros  $\Delta U$ . En el caso de restricción variable manipulada:

$$\begin{bmatrix} u(k_i) \\ u(k_i + 1) \\ u(k_i + 2) \\ \vdots \\ u(k_i + N_c - 1) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix} u(k_i - 1) + \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ I & I & I & \cdots & 0 \\ \vdots & & & & \\ I & I & \cdots & I & I \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \\ \vdots \\ \Delta u(k_i + N_c - 1) \end{bmatrix}$$

Reescribiendo en forma compacta, con  $C_1$  y  $C_2$  correspondiendo a las matrices apropiadas, se impone las restricciones de control como:

$$\begin{aligned}-(C_1 u(k_i - 1) + C_2 \Delta U) &\leq -\Delta U^{min} \\ (C_1 u(k_i - 1) + C_2 \Delta U) &\leq \Delta U^{max}\end{aligned}$$

donde  $U^{min}$  y  $U^{max}$  son vectores columna con  $N_c$  elementos de  $u^{min}$  y  $u^{max}$  respectivamente.

Finalmente, la función objetivo que busca minimizar el vector de parámetros  $\Delta U$  es la siguiente:

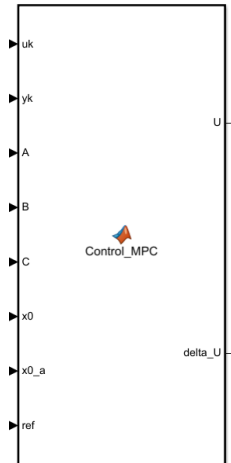
$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U$$

sujeito a las siguientes desigualdad:

$$\begin{bmatrix} -C_2 \\ C_2 \end{bmatrix} \Delta U \leq \begin{bmatrix} -U^{min} + C_1 u(k_i - 1) \\ U^{max} - C_1 u(k_i - 1) \end{bmatrix}$$

## 4.5 Implementación en simulink

Para la implementación en simulink ha sido necesario contar con el identificador recursivo, para obtener el modelo en el espacio de estados. Se ha contruido la siguiente *matlab-function*:



**Figura 4.1** Bloque del MPC.

Como entradas se tienen las siguientes:

- $u_k$ , la acción de control aplicada en el instante actual.
- $y_k$ , la salida del sistema en el instante actual.
- A,B y C, las matrices que forman el modelo en el espacio de estados.
- $x_0$ , variables de estado en el instante actual
- $x_{0a}$ , variables de estado en el instante anterior.
- ref, valor de referencia en el que se quiere controlar la salida.

Como salidas se tienen las siguientes:

- $U$ , acción de control a aplicar.
- $\Delta U$ , variación de la acción de control. Valor que tiende a 0 y sirve para comprobar que el controlador está optimizando correctamente.

A continuación se muestra el código de la función.

#### Código 4.1 MPC.

```
function [U,delta_U] = Control_MPC(uk,yk,A,B,C,x0,x0_a,ref)
coder.extrinsic('opt_problem');
N=4;

nx=size(A,2); % number of states
nu=size(B,2); % number of inputs

%Espacio de estados aumentado
A_a=[A [0;0];C*A 1];
B_a=[B;C*B];
C_a=[0 0 1];

F=zeros(N,3);
Matriz_aux=zeros(1,3);
for i=1:N
Matriz_aux=C_a*A_a^i;
F(i,1)=Matriz_aux(1,1);
F(i,2)=Matriz_aux(1,2);
F(i,3)=Matriz_aux(1,3);
end

Phi=zeros(N,N);
for i=1:N
for j=1:i
Phi(i,j)=C_a*A_a^(i-j)*B_a;
end
end

xk=[x0(1)-x0_a(1);x0(2)-x0_a(2);yk];
R=100;
Q=1;
W=ones(N,1)*ref; % reference
% Build R_hat
R_hat = kron(eye(N),R);
Q_hat = kron(eye(N),Q);

%Restricciones
Umin=0.1;
Umax=5;
```

```

Au=[-1;1];
bu=[-Umin+uk;Umax-uk];

Au_hat=kron(eye(N),Au);
bu_hat=kron(ones(N,1),bu);

H=Phi'*Q_hat*Phi+Q_hat;
f=(F*xk-W)'*Q_hat*Phi;

%Salida 0<Y<50 Ax*Y<=bx
%Entrada 0<U<5 Au*U
% UMPC=zeros(N,1);
% UMPC=inv(Phi'*Phi+R_hat)*(Phi'*W-Phi'*F*xk);
% U=UMPC(1)+uk;
% delta_U=UMPC(1);

UMPC=zeros(N,1);
UMPC=opt_problem(H,f,Au_hat,bu_hat);
U=UMPC(1)+uk;
delta_U=UMPC(1);

```

Cabe destacar que la acción de control se optimiza usando la función *"quadprog"* de matlab. Para usar esta función en simulink es necesario llamar a otra función creada como es *"opt\_problem"*, y esta función es la que ejecuta *"quadprog"*.

---

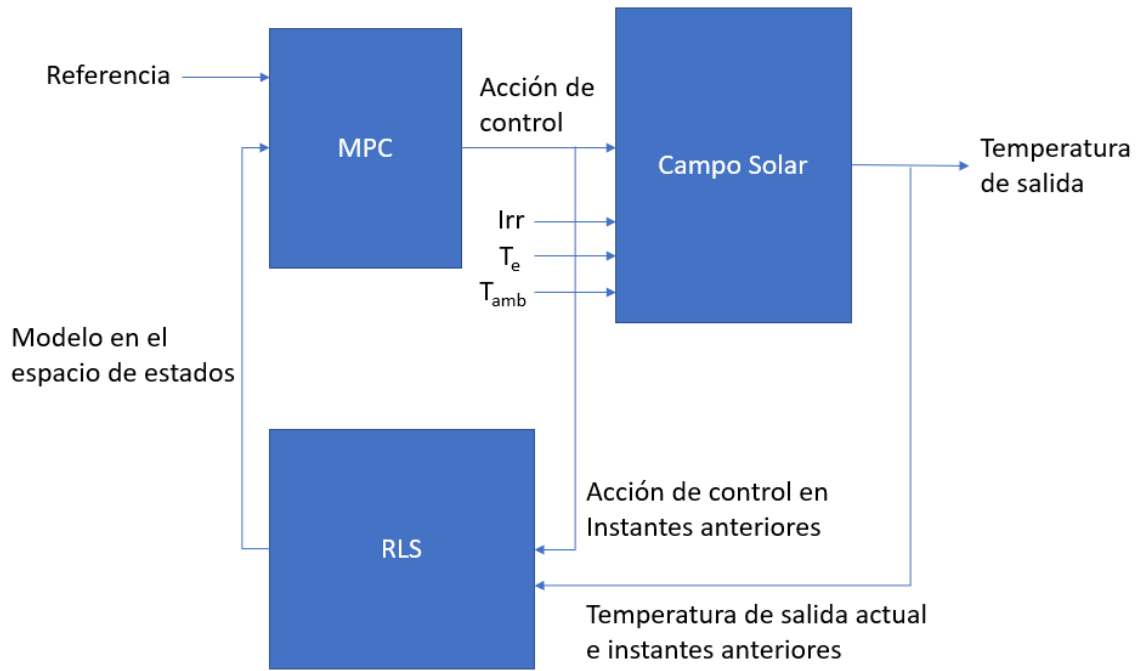
**Código 4.2** Función para ejecutar *quadprog*.

```

function [delta_U] = opt_problem(H,F,A,b)
delta_U=quadprog(H,F,A,b);
end

```

El diagrama de bloques usado a la hora de implementar el controlador es el siguiente:



**Figura 4.2** Diagrama de bloques.

Como se puede ver el campo solar tiene cuatro entradas de las cuales sólo el caudal de entrada es la variable manipulable. El resto de entradas son perturbaciones: irradiancia, temperatura ambiente y temperatura de entrada. El bloque RLS (Recursive least squares) es el sistema de identificación ya descrito anteriormente. Recibe como argumentos la acción de control en los instantes anteriores y la temperatura de salida del instante actual y los instantes anteriores. Este bloque se encarga también de construir el modelo en el espacio de estados. Para ello se identifica un modelo de primer orden de la forma de (4.10)

$$y(k) = -\theta_1 y(k-1) + \theta_2 u(k-1) \quad (4.10)$$

Esta ecuación hay que ponerla en el espacio de estados:

$$\begin{aligned} \dot{x}(k+1) &= A \cdot x(k) + B \cdot u(k) \\ y(k) &= C \cdot x(k) \end{aligned}$$

Para ello se elige como vector de estados

$$x(k) = \begin{bmatrix} y(k-1) \\ u(k-1) \end{bmatrix}$$

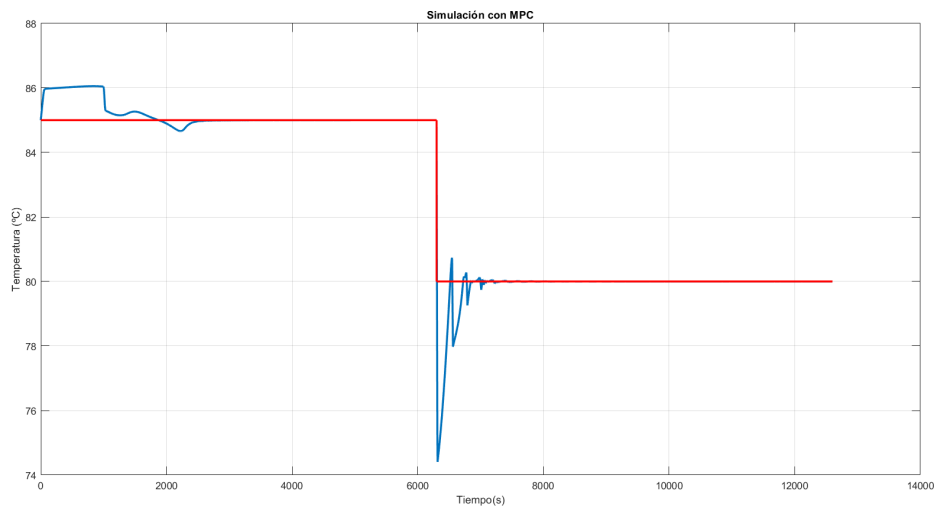
Con tal vector de estados se forma el sistema

$$A = \begin{bmatrix} -\theta_1 & \theta_2 \\ 1 & 0 \end{bmatrix}$$

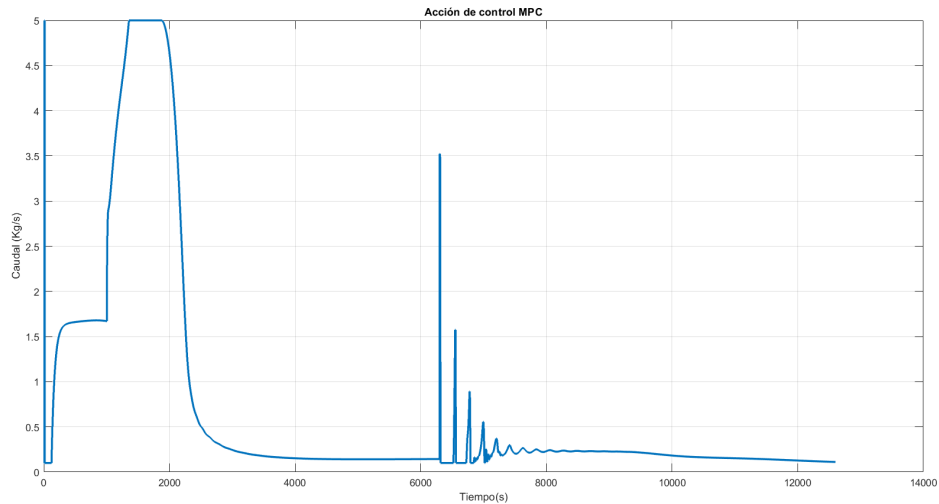
$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = [-\theta_1 \quad \theta_2]$$

En la simulación realizada se ha dejado un tiempo de 1000 segundos sin que interviniese el controlador para que el sistema se estabilizara. A partir de ese instante, con una entrada escalón en la referencia se ha probado el controlador MPC. En la Figura 4.3 se pueden ver los resultados.



**Figura 4.3** Resultados del MPC.



**Figura 4.4** Acción de control del MPC.

Como se ve el controlador controla en la temperatura deseada, pero se producen grandes sobreoscilaciones.

Para hallar el horizonte de predicción se han realizado varias pruebas, comparando un índice de error. Éste índice de error es la suma de las diferencias entre la referencia y la salida al cuadrado en todo el tiempo de control.

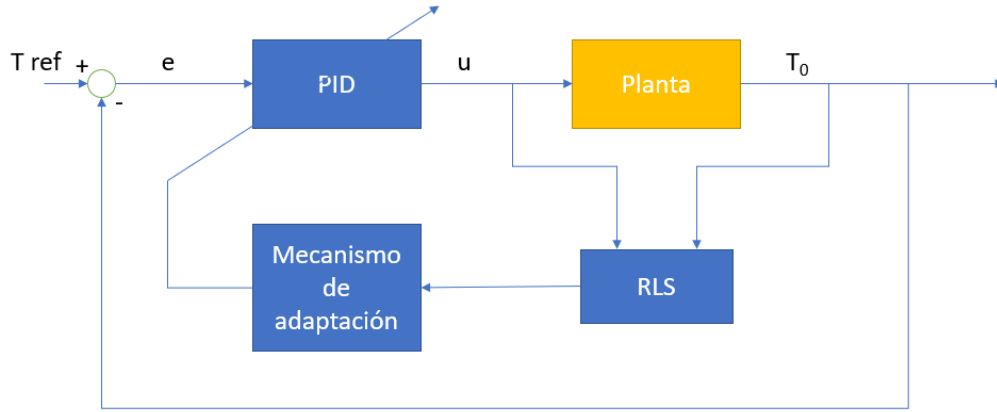
Se ha obtenido que el mejor horizonte de predicción ha sido el de  $N = 4$ .

Para intentar mejorar la sobreoscilación se probó cambiar con menos frecuencia el modelo, dejándolo fijo en algunos puntos de funcionamiento, pero se obtenían peores resultados.

## 4.6 PID adaptativo

Para mejorar los resultados obtenidos se ha probado a usar un PID adaptativo. El esquema de control sería el siguiente:





**Figura 4.5** Esquema del PID adaptativo.

Para el cálculo de los parámetros se hará uso de las reglas de Ziegler-Nichols tal y como se explica en [3].

Para ello se considera la siguiente función de transferencia:

$$G(z) = z^{-1} \frac{bz^{-1}}{(1 - az^{-1})}$$

Los parámetros  $a$  y  $b$  son estimados en línea por el algoritmo recursivo. En este caso tendremos:

$$\begin{aligned} a &= -\theta_1 \\ b &= \theta_2 \end{aligned}$$

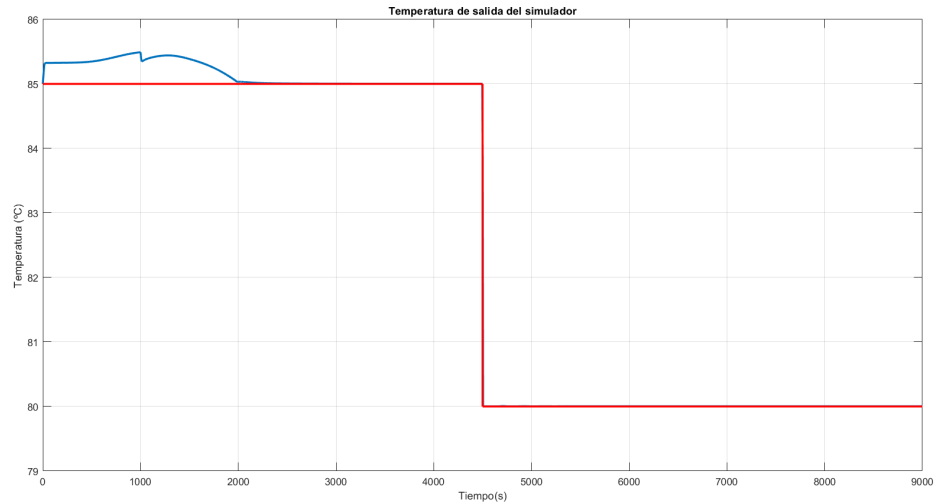
Con estos parámetros se hace el siguiente cálculo:

$$\begin{aligned} \tau &= -T / \log(a) \\ K &= b / (1 - a) \end{aligned}$$

Teniendo la ganancia y la constante de tiempo se calculan los parámetros de la siguiente forma:

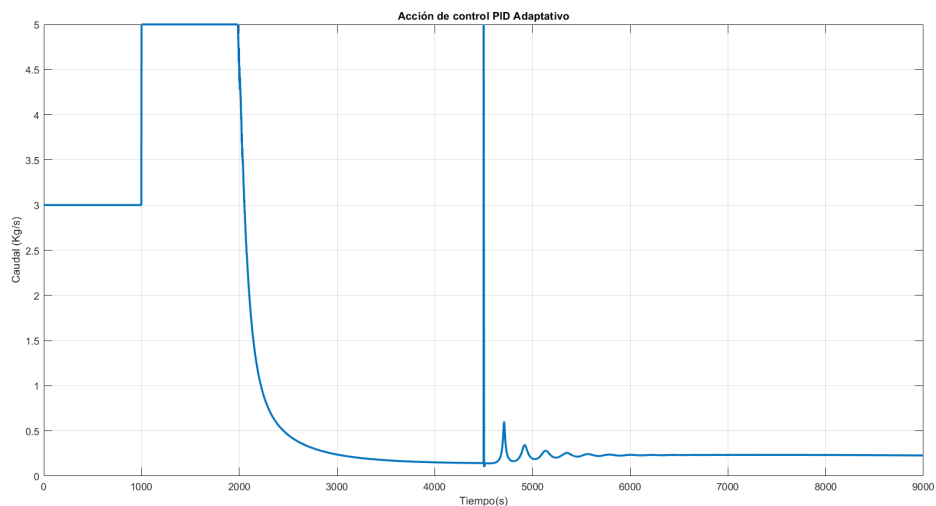
$$\begin{aligned} K_p &= \frac{\alpha 1.2 \tau}{K \tau_d} \\ T_I &= 2 \tau_d \\ T_D &= T_I / 4 \end{aligned}$$

Tras realizar varias simulaciones en Simulink, se comprueba que la parte integral del PID hace que el sistema no controle, llegando a inestabilizar el sistema. Se deja por tanto solo la parte proporcional, consiguiendo muy buenos resultados como se puede ver en Figura 4.6.



**Figura 4.6** Resultados del PID adaptativo.

Al inicio de la simulación se tiene una entrada constante de 3 Kg/s para estabilizar el sistema. A partir de 1000 segundos es cuando empieza a actuar el controlador. La acción de control tiene unos límites de saturación entre 0.1 Kg/s y 5 Kg/s. Realmente la saturación inferior es cero, pero para evitar problemas de cálculo en simulink se toma el valor de 0.1 Kg/s.



**Figura 4.7** Acción de control del PID adaptativo.

# 5 Identificación de fallos

---

*La ciencia es la progresiva aproximación del hombre al mundo real*

MAX PLANCK

Los procesos industriales de hoy en día son cada vez más complejos, conteniendo una gran cantidad de variables. Esta complejidad hace que conforme se produzca algún fallo en los sensores, toda predicción que se pueda hacer del sistema es imprecisa, llegando a ser inservible el modelo obtenido con las técnicas explicadas en el capítulo anterior.

Una de las soluciones que se pueden dar para detectar estos fallos, es mediante redundancia de sensores. Pero esta solución es cara y puede provocar un paro parcial o total de la planta, con la consecuente pérdida económica.

Para ello, en este trabajo se ha implementado un sistema de detección de fallos, basado en redundancia de software, algo que hace que se produzca una detección rápida y sin una gran pérdida económica.

Para la identificación de fallos hay distintas técnicas [5]: la mayoría de ellas están basadas en la evaluación de los residuos que se generan entre un modelo y el sistema como en [7]. Tienen la desventaja de que se necesita un buen modelo del sistema, para de esta manera detectar los fallos.

Hay otras técnicas basadas en inteligencia artificial, por ejemplo empleando redes neuronales se puede predecir si se ha producido un fallo o no. Dependiendo de como se implemente pueden ser técnicas muy robustas, aunque tienen una gran carga computacional y no son sencillas de implementar.

Para este trabajo se ha escogido un sistema de identificación basado en observadores de estados, concretamente en observadores de Luenberger [10].

## 5.1 Conceptos fundamentales de un observador

En determinadas circunstancias resulta necesario estimar el valor de ciertas variables de estado que no son medidas en un sistema. La estimación de este tipo de variables se

denomina observación. Un dispositivo (o programa) que estima u observa las variables de estado se llama observador de estados (o simplemente observador).

A la hora de dar una clasificación de los tipos de observadores, se fija en el número de variables que son observadas. De esta manera, propone una clasificación de observadores en tres tipos:

- **Orden completo:** el observador capta todas las variables del sistema, sin importar si algunas están disponibles para una medición directa.
- **Orden reducido:** el observador estima menos de  $n$  variables de estado, en donde  $n$  es la dimensión del vector de estado.
- **Orden mínimo:** es un observador de orden reducido con el mínimo orden posible, es decir, si  $n$  es la dimensión del vector de estado y  $m$  es la dimensión del vector de salidas, el observador de orden mínimo observa  $n-m$  variables.

Dentro de cada una de estas familias, podemos tener, a su vez, otras subdivisiones. Por ejemplo, si consideramos el tipo de sistema que estamos tratando podemos tener *observadores para estados lineales* y *observadores para sistemas no lineales*.

A continuación, se explicarán los fundamentos de observadores de estados para el caso más general, el de observadores de orden completo para sistemas lineales.

## 5.2 Observadores de Estado para Sistemas Lineales

Consideramos que tenemos un sistema en espacio de estados de la siguiente forma:

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad (5.1)$$

$$y(t) = C \cdot x(t) \quad (5.2)$$

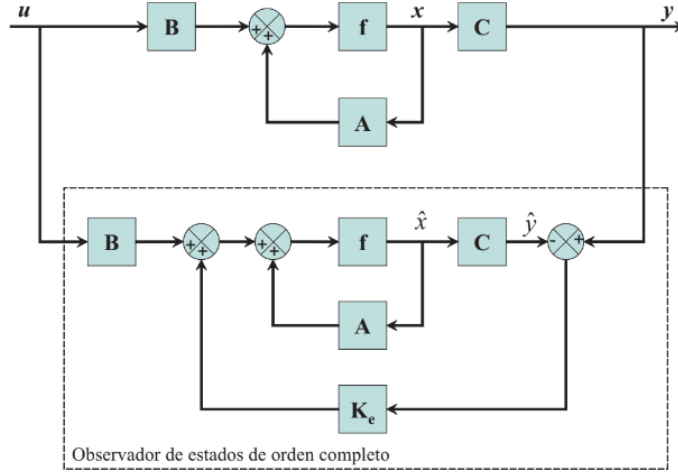
En donde  $x(t)$  es el vector de estado (vector de dimensión  $n$ ),  $u(t)$  es el vector de entradas,  $y(t)$  es el vector de salidas, y  $A$ ,  $B$  y  $C$  son matrices de coeficientes constantes.

Podemos definir un *observador de estados genérico* para dicho sistema como:

$$\dot{\hat{x}}(t) = A \cdot \hat{x}(t) + B \cdot u(t) + K_e \cdot (y(t) - C \cdot \hat{x}) \quad (5.3)$$

$$\hat{y}(t) = C \cdot \hat{x}(t) \quad (5.4)$$

El último término, es un valor de corrección que contiene la diferencia entre la salida medida y la salida estimada. El término de corrección ayuda a reducir los efectos producidos por la diferencia entre el modelo dinámico y el sistema real. En la Figura 5.1 se puede ver de manera esquemática el sistema en espacio de estado y su observador genérico correspondiente.



**Figura 5.1** Esquema del observador.

Podemos obtener la dinámica del error en la estimación del estado restando la ecuación (5.1) de la ecuación (5.3):

$$\begin{aligned}\dot{x}(t) - \dot{\hat{x}}(t) &= A \cdot x(t) - A \cdot \hat{x}(t) - K_e \cdot (C \cdot x(t) - C \cdot \hat{x}(t)) \\ &= (A - K_e C) \cdot (x(t) - \hat{x}(t))\end{aligned}$$

Y definiendo la diferencia entre  $x(t)$  y  $\hat{x}(t)$  como el vector de error  $e_x(t)$

$$e_x(t) = x(t) - \hat{x}(t)$$

Entonces el error en la estimación del estado, nos queda:

$$\dot{e}_x(t) = (A - K_e C) \cdot e_x(t) \quad (5.5)$$

A partir de la ecuación (5.5) podemos ver que el comportamiento dinámico del vector de error se determina mediante los valores característicos de la matriz  $(A - K_e C)$ . Si la matriz es estable, el vector convergerá a cero para cualquier vector de error inicial ( $e(0)$ ). Es decir que  $\hat{x}(t)$  convergerá a  $x(t)$  sin considerar los valores de  $\hat{x}(0)$  y  $x(0)$ . Si se eligen los valores característicos de la matriz  $A - K_e C$  de manera que el comportamiento dinámico del vector de error sea asintóticamente estable y suficientemente rápido, cualquier vector de error tenderá a 0 con una velocidad adecuada.

Al igual que para el error en la estimación del estado, podemos obtener el error en la estimación de la salida como:

$$e_y(t) = y(t) - C \cdot \hat{x}(t)$$

Que está directamente relacionado con el error en la estimación del estado de la siguiente manera:

$$e_y(t) = C \cdot e_x(t)$$

### 5.3 Observadores de estado para Detección y Diagnóstico de estados

El objetivo es identificar y aislar los fallos que se produzcan en el campo de colectores. Los fallos se simularán con un offset en la temperatura en un momento dado.

Los sensores que estarán sometido bajo estudio serán el de temperatura de salida del campo de colectores, el de temperatura de entrada en el campo de colectores, el de irradiancia, el de temperatura ambiente y el del caudal de entrada.

Como se vio anteriormente, el observador lo que hace es ajustar el error entre el modelo con el sistema real mediante una ganancia, convergiendo el error a 0. Si se produce un fallo, habrá un cierto instante en el que el error deje de converger y tomará un valor muy distinto a 0.

Para el aislamiento de fallos la técnica que se empleará será la de utilizar un banco de observadores. Debido a la flexibilidad que nos aporta el algoritmo RLS, se puede construir un modelo entre variables que estén estrechamente relacionadas. Gracias a esto, nos aprovechamos para poder aislar todos los fallos.

Los modelos construidos son:

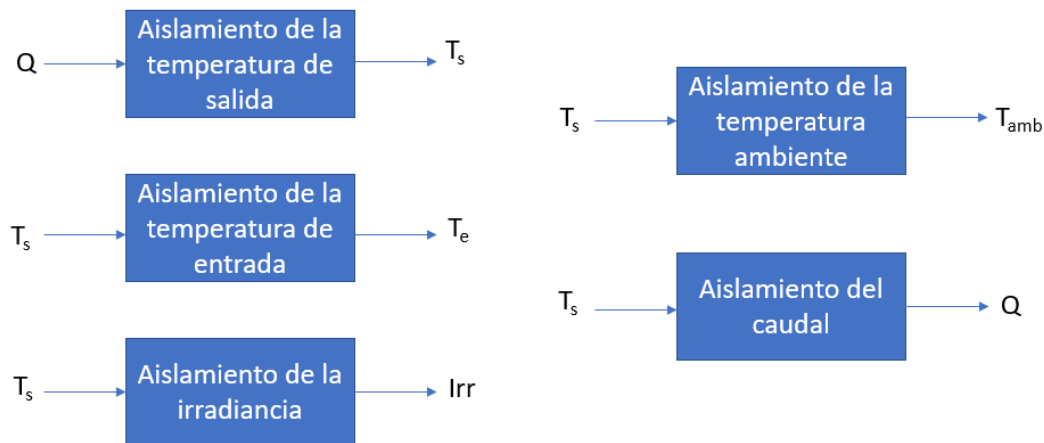
- Modelo que relaciona el caudal con la temperatura de salida.
- Modelo que relaciona la temperatura de salida con la temperatura de entrada.
- Modelo que relaciona la irradiancia con la temperatura de salida.
- Modelo que relaciona la temperatura ambiente con la temperatura de salida.
- Modelo que relaciona el caudal con la temperatura de salida.

Todos los modelos tienen como salida la variable de la que se quiere detectar el fallo. La temperatura de salida actúa generalmente como entrada del sistema, debido a que está relacionada directamente con todas las magnitudes. Al producirse un fallo en algún sensor provocará un cambio importante en los parámetros del modelo. Todos los modelos son de la forma:

$$y(k) = -\theta_1 y(k-1) - \theta_2 y(k-2) + \theta_3 u(k-1) + \theta_4 u(k-2)$$

Siendo  $y$  la variable de salida del modelo y  $u$  la variable de entrada.

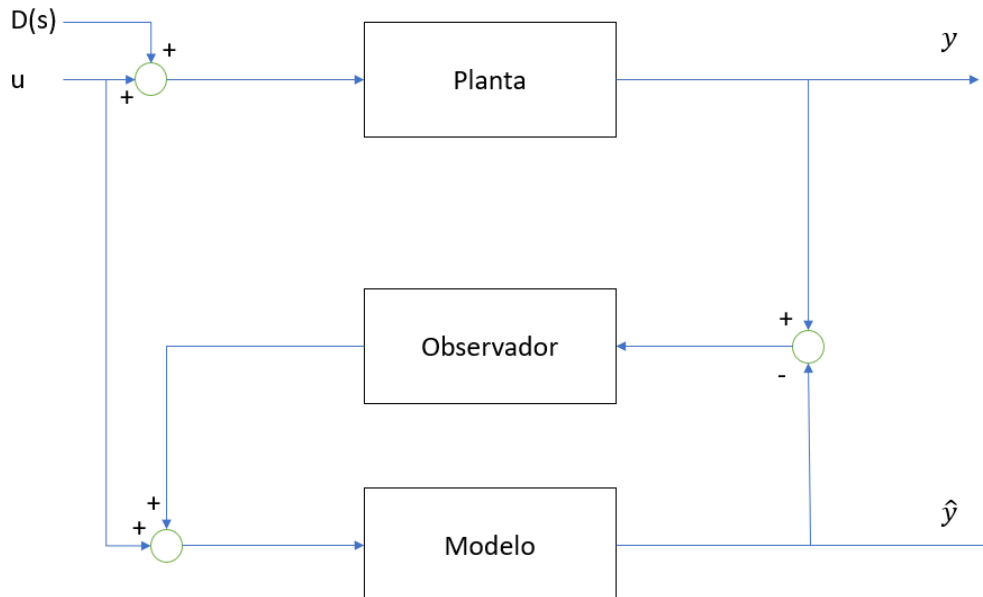
$\theta_1, \theta_2, \theta_3$  y  $\theta_4$  son los parámetros del modelo que cambiarán bruscamente cuando se produzca un fallo.



**Figura 5.2** Banco de observadores.

Aprovechando el modelo que se obtenía por el algoritmo de mínimos cuadrados recursivos, se obtiene el modelo en el espacio de estados para poder así implementar el observador de Luenberger.

Se ha implementado el siguiente diagrama de bloques en simulink:



**Figura 5.3** Diagrama de bloques.

El bloque *modelo* consiste en el sistema identificado por RLS (Recursive Least Square). Para su obtención se aplica lo siguiente:

El modelo obtenido es de la forma

$$y(k) = -\theta_1 y(k-1) - \theta_2 y(k-2) + \theta_3 u(k-1) + \theta_4 u(k-2)$$

Esta ecuación hay que ponerla en el espacio de estados:

$$\begin{aligned}\dot{x}(k+1) &= A \cdot x(k) + B \cdot u(k) \\ y(k) &= C \cdot x(k)\end{aligned}$$

Para ello se elige como vector de estados

$$x(k) = \begin{bmatrix} y(k-1) \\ y(k-2) \\ u(k-1) \\ u(k-2) \end{bmatrix}$$

Con tal vector de estados se forma el sistema

$$A = \begin{bmatrix} -\theta_1 & -\theta_2 & \theta_3 & \theta_4 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = [-\theta_1 \quad -\theta_2 \quad \theta_3 \quad \theta_4]$$

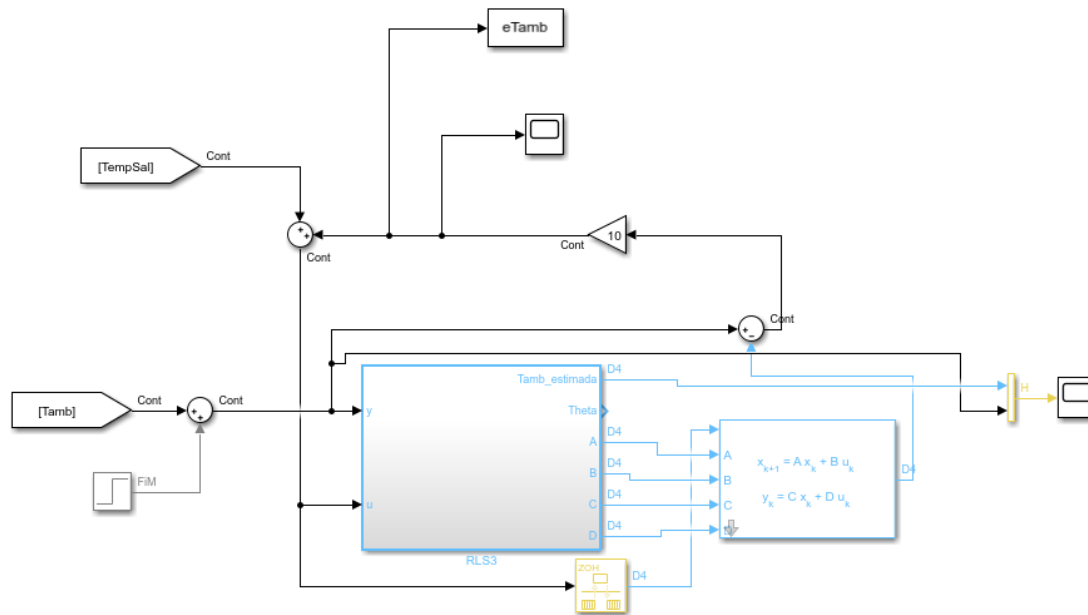
El bloque del observador va a consistir en una ganancia. Su valor se elige heurísticamente teniendo presente que el sistema sea estable y permita detectar el fallo sin ningún tipo de problema. Se elige una ganancia de 10.

La planta será el campo de colectores del simulador, consistente en ecuaciones no lineales. La variable  $u$  será el caudal de entrada, con  $D(s)$  se representan las perturbaciones que recibe el sistema, y es la temperatura de salida del campo de colectores e  $\hat{y}$  es la temperatura de salida estimada por el modelo.

El funcionamiento consiste en comparar la diferencia de la temperatura de salida real con la obtenida por el modelo y corregir esa diferencia con la ganancia del observador.

Dicho todo lo anterior, se muestra un ejemplo de cómo se ha implementado en simulink para el caso de temperatura ambiente.





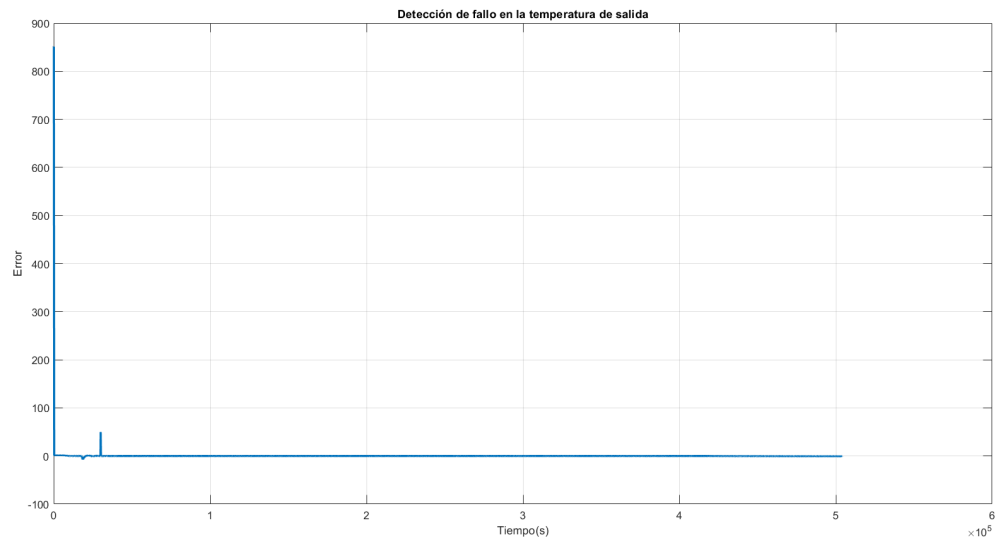
**Figura 5.4** Implementación en Simulink.

En la Figura 5.4 se muestra cómo lo que equivaldría al modelo en la Figura 5.3 está representado con otro tiempo de muestreo, concretamente de 5 segundos. El tiempo de muestreo es muy importante, tiene que ser lo suficientemente grande para que se puedan detectar los errores. A continuación, se realiza una simulación provocando errores simultáneamente para comprobar que el sistema detecta y aísla los fallos. Todos los fallos han consistido en un offset de 5.

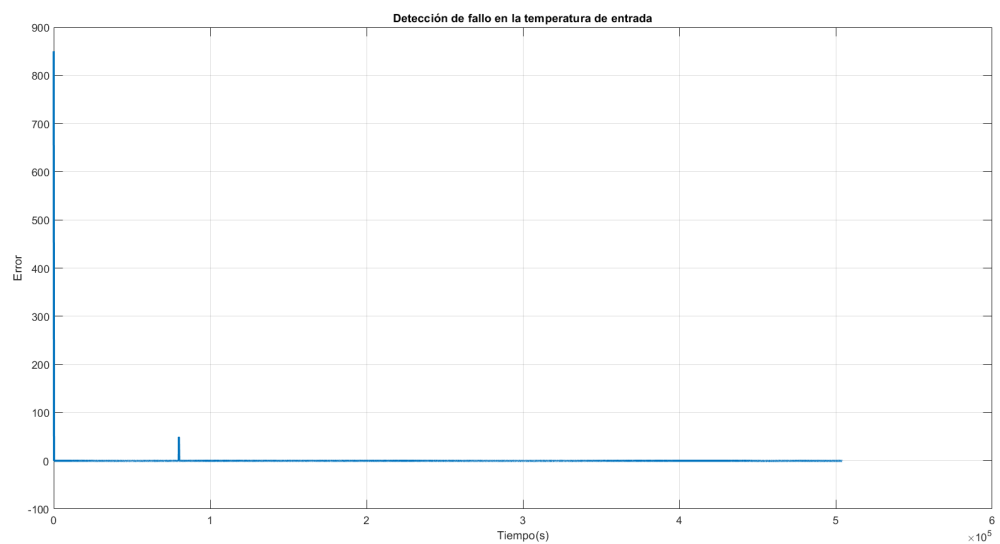
**Tabla 5.1** Tiempo en que se producen los fallos.

Sensor	Tiempo de fallo
Temperatura de salida	1500 s
Temperatura de entrada	4000 s
Irradiancia	1000 s
Temperatura ambiente	3000 s
Caudal	2000 s

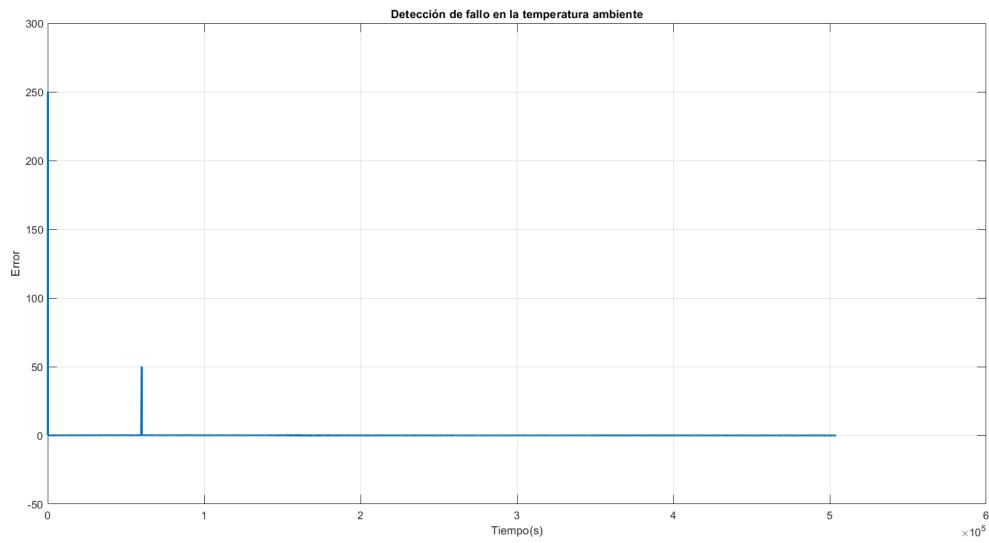
Se pasa a mostrar los resultados



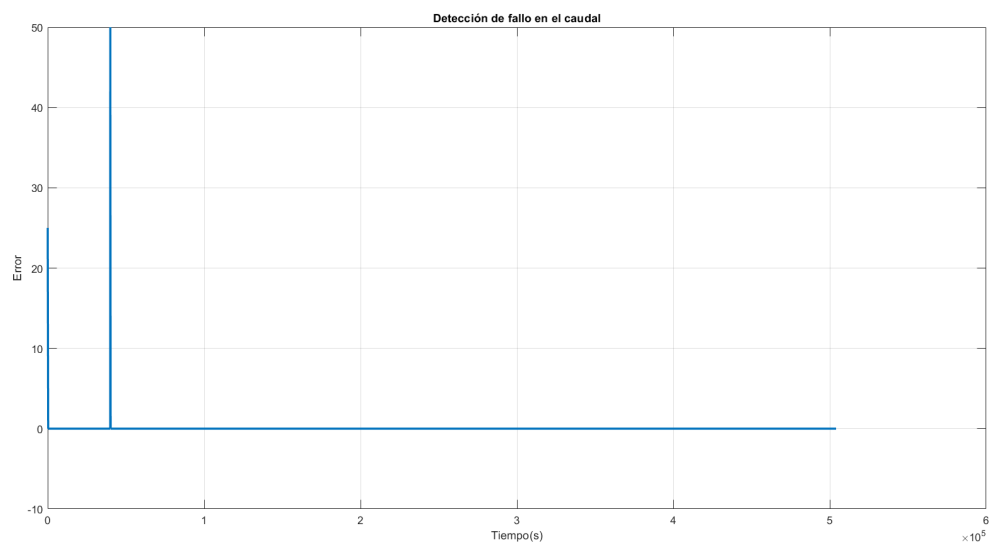
**Figura 5.5** Fallo en la temperatura de salida.



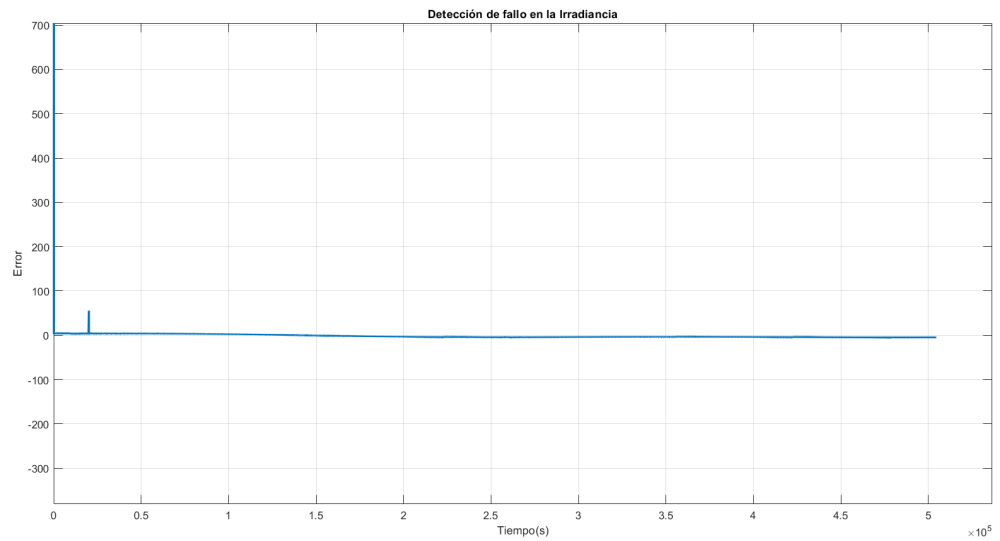
**Figura 5.6** Fallo en la temperatura de entrada.



**Figura 5.7** Fallo en la temperatura ambiente.



**Figura 5.8** Fallo en el caudal.



**Figura 5.9** Fallo en la irradiancia.

Como se puede observar en todas las gráficas, en los instantes en los que se producen los fallos el error se aleja de cero y se produce un pulso. Rápidamente el sistema se adapta y vuelve a converger a cero. Todos los fallos provocados en la simulación son detectados y aislados sin observar ninguna interferencia.

## 6 Control tolerante a fallos

*Cuando una puerta se cierra, otra se abre, pero a menudo vemos tanto tiempo y con tanta tristeza la puerta que se cierra que no notamos otra que se ha abierto para nosotros.*

ALEXANDER GRAHAM BELL

En este capítulo se hará una demostración de como funcionaría el sistema de detección de fallos. Los fallos serán simulados mediante offsets de 5. Se han simulado fallos tanto en la temperatura de salida del campo de colectores como en el caudal.

Durante la implementación del sistema surgieron varios problemas. Uno de ellos es que al estar controlando el sistema y cambiar la referencia, se detectaba como fallo. Otro problema que se detectó fue que, al inicio de la simulación se usaba un caudal de entrada constante para estabilizar el sistema. A partir de un cierto tiempo, al sistema se le introducía una referencia y se producía también un gran offset que detectaba como un fallo.

Por tanto, para paliar estos casos se ha hecho una función de postprocesado:

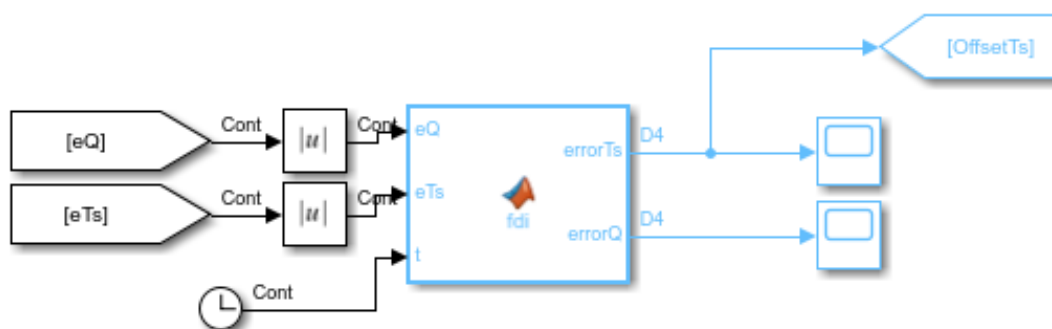


Figura 6.1 Función para procesar fallos.

Cuando se produce un cambio de referencia, se detecta que ha habido un fallo tanto en el caudal como en la temperatura de salida, así pues se considera que si se producen dos fallos simultáneamente, no hay fallo.

Para la detección de un cambio brusco como al inicio del control se puede considerar lo mismo. También lo que se hace es que el fallo se considera a partir de un cierto umbral, por lo que los cambios bruscos pequeños se filtran también.

---

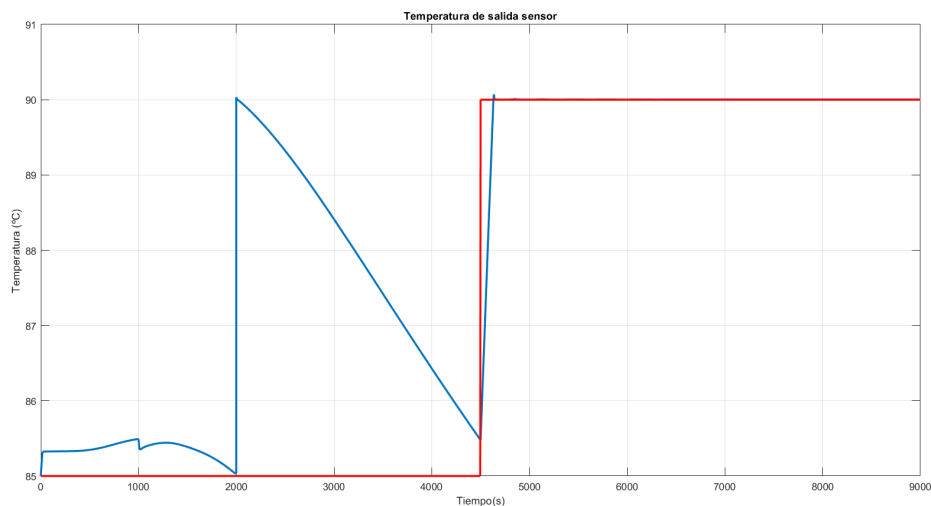
**Código 6.1** Clasificador de fallos.

```
function [errorTs,errorQ] = fdi(eQ,eTs,t)

if t>50
if eQ>30 && eTs>30
errorTs=0;
errorQ=0;
elseif eQ>30 && eTs<30
errorQ=1;
errorTs=0;
elseif eQ<30 && eTs>30
errorTs=1;
errorQ=0;
else
errorTs=0;
errorQ=0;
end
else
errorTs=0;
errorQ=0;
end
```

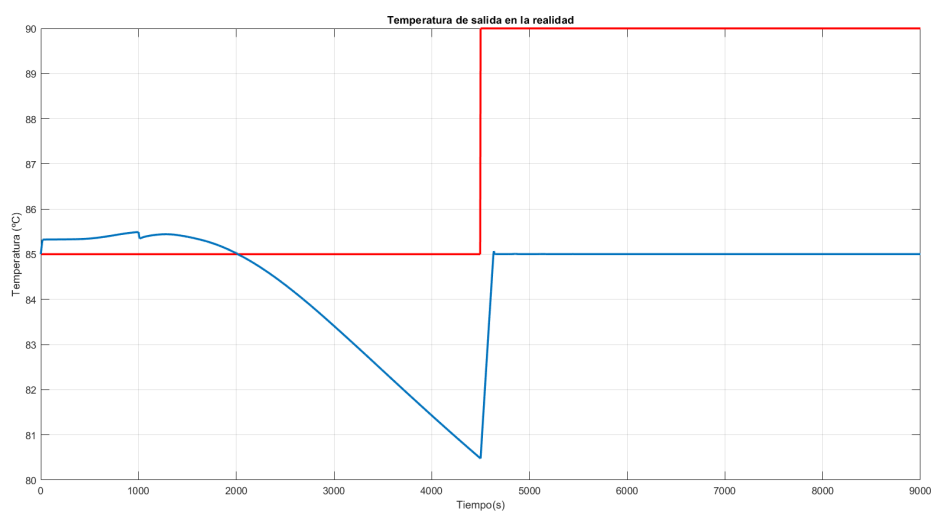
Como se puede ver en el código, el identificador de fallos comienza a funcionar a partir de un cierto tiempo que el error esté estabilizado en torno al 0.

Se va a mostrar cómo funcionaría el sistema con un fallo en la temperatura de salida.



**Figura 6.2** Fallo en la medida del sensor.

En la Figura 6.2 se puede ver que a partir del instante  $t=2000s$  se produce un fallo en el sensor que mide la temperatura de salida. A pesar del fallo, la acción de control hace que el sistema evolucione hasta la referencia según la medida del sensor, pero lo que realmente ocurre es lo que se puede ver en Figura 6.3 donde se observa un desfase de  $5^{\circ}C$ .



**Figura 6.3** Temperatura de salida real.

Para solucionarlo hay que corregir la medida del sensor cuando se detecte un fallo. Para ello se añade un bloque que rectifique dicho offset al producirse un fallo.

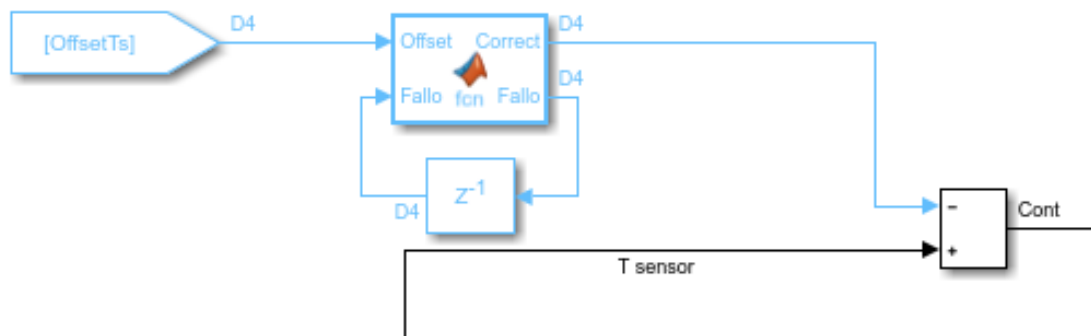


Figura 6.4 Corrección del fallo de temperatura.

**Código 6.2** Corrección del fallo.

```
function [Correct,Fallo] = fcn(Offset,Fallo)

if Offset == 1 || Fallo == 1
Correct=5;
Fallo=1;
else
Fallo=0;
Correct=0;
end
```

Al incluir dicha corrección, el sistema durante la simulación detecta el fallo y corrige el offset producido por el sensor, controlando en la temperatura de referencia.

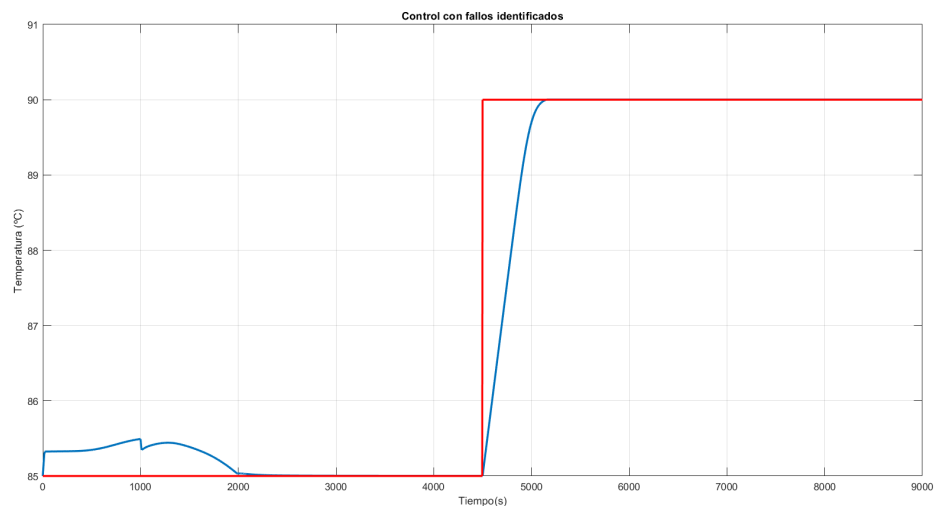
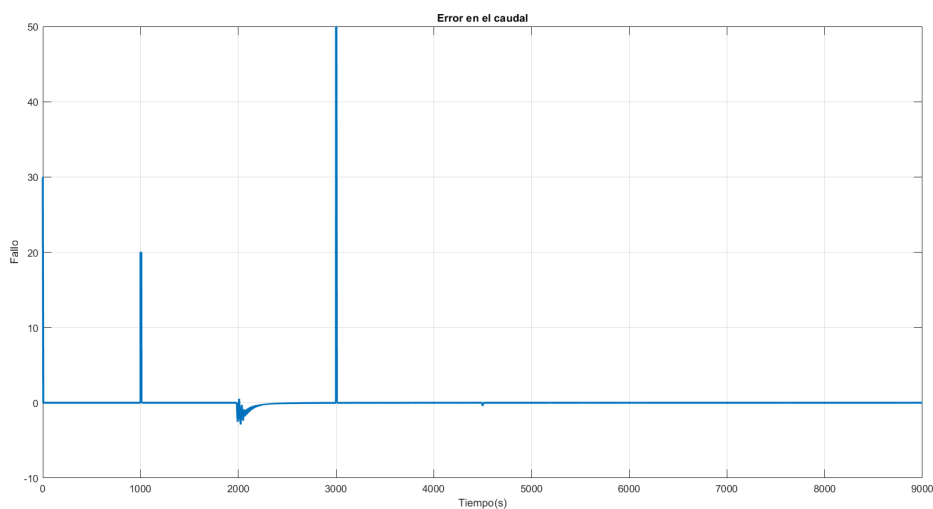


Figura 6.5 Control con fallos identificados.

Se va a mostrar como ha funcionado el postprocesado. En la Figura 6.6 se muestra como

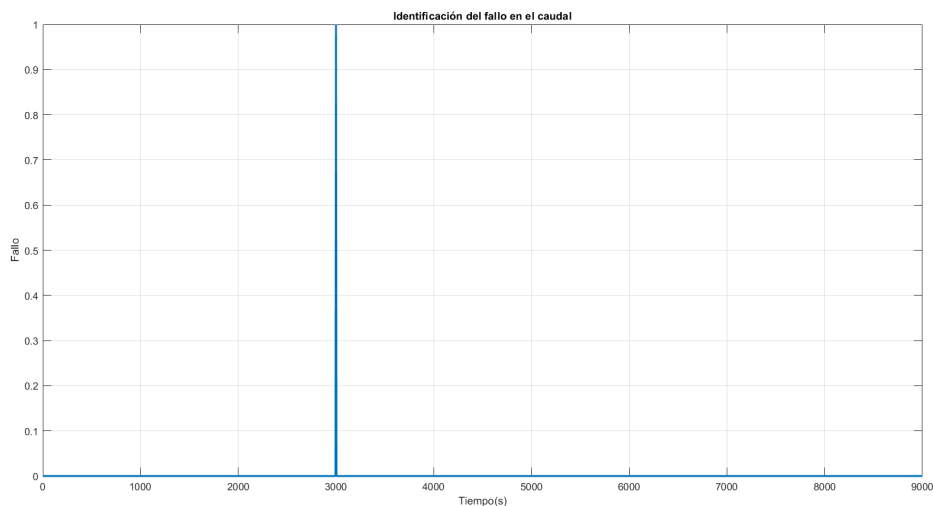


el sistema identificaría dos fallos, el primero a los 1000 segundos y el segundo a los 3000 segundos.



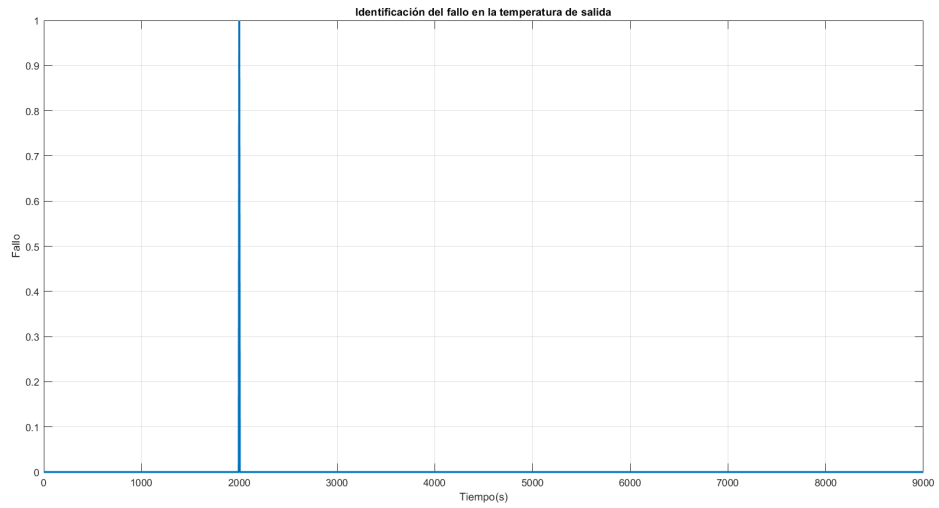
**Figura 6.6** Error en el caudal.

Pero el primer fallo que detecta no es realmente un fallo, sino que se trata de que a partir de ese instante se pone el sistema a controlar. Mediante el postprocesado se filtran estos problemas.



**Figura 6.7** Fallo en el sensor del caudal.

Por último se muestra como se detectó también el fallo en la temperatura de salida.



**Figura 6.8** Fallo en el sensor de la temperatura de salida.

## 6.1 Conclusiones

Se ha demostrado como se puede construir un sistema de detección de fallos usando modelos y control adaptativos. Esto hace que se simplifique mucho el proceso de implementación, al no tener que recopilar datos de los distintos sistemas, verificar que el modelo es correcto y preocuparse de que el modelo siga siendo válido durante el tiempo.

Con la identificación en línea que se ha realizado, se evita la creación de modelos y su posterior identificación, no teniendo que tener una gran cantidad de datos para validar los modelos desarrollados. Además, con el paso del tiempo estos modelos dejarán de representar fielmente el sistema modelado, debido a cambios en los parámetros de las ecuaciones, debido principalmente al envejecimiento del mismo.

Viendo los resultados obtenidos en el control de la planta solar, se puede decir que el empleo de un control adaptativo es una gran opción para ser combinado con el sistema de identificación realizado. El uso de controladores predictivos basados en modelos, aunque tienen un buen control, no es tan fino como el adaptativo, siendo mejor usarlos con modelos fijos, que no cambien con el tiempo. El hecho de que los parámetros del modelo cambien con tanta frecuencia hace complicado la predicción a largo plazo de la acción de control.

El sistema de detección de fallos aquí explicado se podría usar sin problemas en sistemas con una dinámica lenta, en el que los valores que se midan cambien lentamente con el tiempo, como es el caso de una planta solar, consiguiendo un control robusto y fiable del sistema.



# Índice de Figuras

---

1.1	Simulador de la planta solar	1
1.2	Esquema del simulador de la planta solar	2
2.1	Esquema de la planta	4
2.2	Campo Solar	5
2.3	Esquema del acumulador	6
2.4	Modelo de parámetros distribuidos del acumulador	7
2.5	Balance de potencia en el generador	10
2.6	Válvula de tres vías	12
3.1	Esquema del algoritmo	20
3.2	Entradas y salidas del modelo no lineal	20
3.3	Entradas y salidas del modelo lineal	21
3.4	Entradas y salidas del algoritmo de identificación	22
3.5	Diagrama de bloques de la simulación realizada	23
3.6	Comparación del modelo obtenido con el simulador	23
3.7	Comparación del modelo obtenido con el simulador	24
3.8	Caudal de entrada variable	24
3.9	Parámetros obtenidos	25
4.1	Bloque del MPC	33
4.2	Diagrama de bloques	36
4.3	Resultados del MPC	37
4.4	Acción de control del MPC	38
4.5	Esquema del PID adaptativo	39
4.6	Resultados del PID adaptativo	40
4.7	Acción de control del PID adaptativo	40
5.1	Esquema del observador	43
5.2	Banco de observadores	45
5.3	Diagrama de bloques	45
5.4	Implementación en Simulink	47
5.5	Fallo en la temperatura de salida	48
5.6	Fallo en la temperatura de entrada	48

5.7	Fallo en la temperatura ambiente	49
5.8	Fallo en el caudal	49
5.9	Fallo en la irradiancia	50
6.1	Función para procesar fallos	51
6.2	Fallo en la medida del sensor	53
6.3	Temperatura de salida real	53
6.4	Corrección del fallo de temperatura	54
6.5	Control con fallos identificados	54
6.6	Error en el caudal	55
6.7	Fallo en el sensor del caudal	55
6.8	Fallo en el sensor de la temperatura de salida	56

# Índice de Tablas

---

2.1	Nomenclatura del modelo del colector	6
2.2	Nomenclatura del acumulador	9
2.3	Nomenclatura del generador	11
5.1	Tiempo en que se producen los fallos	47





# Índice de Códigos

---

3.1	Algoritmo RLS	22
4.1	MPC	34
4.2	Función para ejecutar <i>quadprog</i>	35
6.1	Clasificador de fallos	52
6.2	Corrección del fallo	54



# Bibliografía

---

- [1] Jesús David Gordón Ballester, *Automatización de planta solar de refrigeración por absorción*, 2016.
- [2] Jesús Agustín Martínez Bravo, *Control tolerante a fallos de planta solar de refrigeración*, 2017.
- [3] Eduardo F. Camacho, Manuel Berenguel, and Francisco R. Rubio, *Advanced control of solar plants*, Dec 1997.
- [4] E.F Camacho and C. Bordons, *Model predictive control*, Springer, 2007.
- [5] Rafael Martinez Guerra and Juan Luis Mata Machuca, *Fault detection and diagnosis in nonlinear systems*, Springer, 2014.
- [6] Daniel Rodríguez Ramírez and Carlos Bordons, *Apuntes de ingeniería de control*, Universidad de Sevilla, 2007.
- [7] Manuel G. Satué and Francisco R. Rubio, *Detección e identificación de fallos en planta solar de refrigeración por absorción*, 2018.
- [8] Liuping Wang, *Model predictive control system design and implementation using matlab*, Springer, 2009.
- [9] F.R. Rubio y M.J. López, *Control adaptativo y robusto*, Universidad de Sevilla, 1996.
- [10] Miguel Angel Niño Zambrano, *Observadores de estados*, 1<sup>a</sup>, 2007.



